An End-to-End Security Architecture to Collect, Process and Share Wearable Medical Device Data

Kurt Rohloff and Yuriy Polyakov
School of Computer Science
New Jersey Institute of Technology
Newark, NJ 07102
Email: {rohloff,polyakov}@njit.edu

***Abstract*—Embedded medical devices, such as wearable devices, are becoming increasingly common, but data from these devices is both very private and highly vulnerable to theft. Data needs to be collected from multiple devices to improve the effectiveness of treatment. The medical devices, data processing sites and intended care givers are often geographically distributed, and operate on different time scales with collected data being aggregated for days or months before analysis and usage. Current approaches to data security do not provide a framework for end-to-end protection, where data can always be encrypted but still used effectively. We present a security architecture with end-to-end encryption that supports 1) secure collection of data from embedded medical devices, 2) protected computing on this data in low-cost commodity cloud environment and 3) restricts the delegation of access to this data to designated recipients. The basis of the architecture comes from recent advances in lattice encryption technologies. This approach leverages recent breakthroughs in Homomorphic Encryption (HE) and Proxy Re-Encryption (PRE) that would practically support specific data aggregation, processing and distribution needs of a secure medical data architecture. This architecture lowers health care data system costs by securely outsourcing computation to cloud computing environments while simultaneously reducing vulnerabilities to some of the most problematic security challenges such as insider attacks and enables additional cost savings with lower-cost embedded medical devices.**

I. Introduction

Irrespective of regulation [1], medical data security, especially data from wearable devices, is a pressing concern. These devices collect the most sensitive data about patients' health day-to-day, often with information about the patients' physical location and well being. The data also needs to be shared and manipulated to be useful. The data is often collected in a different location from where it is processed and the processing location is often different from where health care providers will interpret the results of processing. These data collection, processing and interpretation steps may often occur over different time scales with data sometimes being collected every few minutes or even seconds in the case of detecting life-threatening emergency alerts. Conversely, wearable medical devices may collect information over days or weeks that is used for diagnostic purposes to decide on courses of treatment.

The pressing security concerns, coupled geographically and temporally distributed data lifecycles, create challenges in creating effective but secure and lower-cost information architectures. It can be difficult to balance security, effectiveness and cost concerns as they are not necessarily monolithic and tradeoffs between them are not always smooth. Current solutions might protect data at times using encryption technologies when data is in motion (when transmitted between storage points) or at rest (when stored). Any substantive manipulation of the data, up to now, has required the data to unencrypted. This means that medical data processing could, until now, only be done in trusted computing environments which are often expensive to set up and maintain because they need to be exclusively managed by highly trusted individuals, often in dedicated facilities. Similarly, because encryption is used to protect data when transmitted point-to-point, data is encrypted only when the intended recipient of the data has been pre-approved. There has been no way for medical data to be encrypted at collection, and then routed later to intended without prior coordination of encryption keys. This need for prior coordination between encryption and decryption points presents another possible set of vulnerabilities where data thieves could have more opportunities to steal decryption keys. Security and effectiveness trade-offs have preventing the widespread use of low-cost cloud computing environments [2] and large engineering effort has been needed to validate the security of wearable medical devices that more easily integrate with a larger security ecosystems [3]. All of these concerns substantially raise the cost of treatment and reduce flexibility.

Based on this problem context, the contributions of this paper are:

- A security model and concrete set of features that should be used in reasoning about security architectures for wearable medical devices.
- An end-to-end architecture for the secure collection, processing and distribution of wearable medical devices data based on recent theoretical and experimental results.
- Motivation for how this architecture enables the secure use of cost-effective cloud computing environments and lowers the costs of wearable medical devices.

Sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under Contract No. FA8750-11-C-0098. The views expressed are those of the authors and do not necessarily reflect the official policy or position of the Department of Defense or the U.S. Government. Distribution Statement "A" (Approved for Public Release, Distribution Unlimited.)

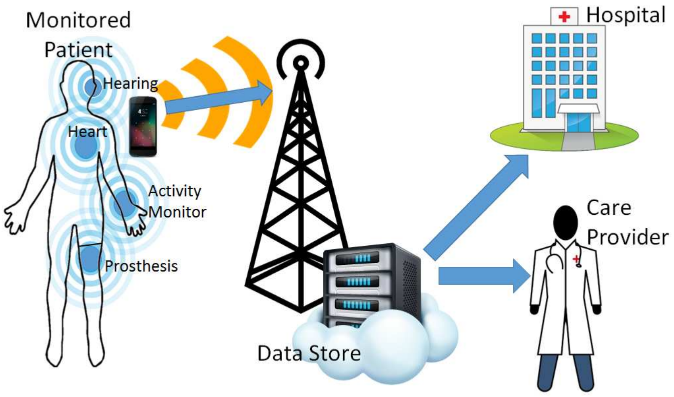

Fig. 1. Architecture USe-Case Overview

The paper is organized as follows. In Section II we identify the security model for the potential vulnerabilities we attempt to security against and identify functional, security and cost features that should be considered when evaluating an information security architecture for wearable medical devices. We present our security architecture in Section III. Section IV discusses initial experimental results. Section V discusses related encrypted computing activities and applications. We conclude the paper with a discussion of our insights, resulting cost reduction and future work in Section VI.

II. Security Model and Evaluation Measures

An overview of the use of our proposed secure architecture is seen in Figure 1. In this figure, multiple body sensors send their data to a network access point, notionally a smart phone with data capability. The sensors could be any number of devices such as heart rate monitors, smart prosthesis diagnostics, hearing aids, etc... where data needs to be collected and transmitted either periodically or in an event-driven manner. The smart phone access point aggregates data from multiple sensors as needed to save on bandwidth, and then transmit the data on commodity data networks to a secure data store which hosts the collected data. The data store runs any processing needed on the collected data, such as diagnostic functions to identify leading indicators of health issues. The data is then sent to approved health care providers.

We envision the security architecture instantiated as a kind of middleware, with adaptability in terms of underlying communication protocols and upper-layer applications that need to manipulate the data. The sensors could communicate with the smart phone through any number of possible standard protocols including BAN, Bluetooth, 802.11 or Zigbee. The smart phone and data store could send data over, for example, packet-switched networks such as the Internet, or on cellular data networks, depending on network access. Similarly, the datastore is notionally any server and could reside in a proprietary data-center or in a cloud computing environment.

To set up the environment, point-to-point communication approvals would need to be established, namely that:

- The sensors would need to be paired with the smart phone.
- The smart phone access point would need to be configured with the URL of IP address of the data store.
- The approval for care providers to receive data would need to be received from the patient or the patient's legally approved representative.

Within the framework of standard information security analysis approaches [4], systems are generally analyzed from the perspective of the Confidentiality, Integrity and Availability (CIA) triad. We particularly seek to address vulnerabilities from the Confidentiality perspective, by enabling the data store to 1) process data without access to the unencrypted data or decryption keys and 2) grant access to the encrypted data without being able to grant access to themselves. We address these issues through a proposed architecture that incorporates an end-to-end encryption capability where embedded medical devices encrypt data at the sensor, and data is never accessible unencrypted until it reaches its intended recipient. As such, we design the architecture to address important data leakage threats to better and more key security concerns [4].

Based on the discussed high-level use cases and security goals, we identify several design measures with which to evaluate and reason over our secure information architecture designs tradeoffs. These design measures are general and are usable to address other information security design challenges. We organize these concerns in terms of security, functionality/performance and cost design measures:

1) **Security**
   a) **Non-Interactive**: We want to minimize or eliminate any human interactions which could slow any data sharing, processing or access delegation, or lead to social engineering vulnerabilities.
   b) **Unencrypted Data**: We want to minimize the incidence of unencrypted data, even during data processing if possible.
   c) **Access**: As few (human *or* machine) participants as possible should have any data access or the ability to access decryption keys to decrypt data.
   d) **Encryption Work Factor**: Any use of encryption to secure data should provide an encryption work factor roughly comparable that eliminates any practical opportunity to compromise confidentiality of the encrypted data. For all practical purposes, this means that the computational effort required to break the encryption through brute force methods has a work factor at least as high as AES-128, a current encryption standard.
2) **Functionality/Performance**
   a) **Latency**: The end-to-end delivery of source data to eventual user should provide a workable end-to-end latency. This measure is highly context dependent. For long-term diagnostics this means that the lag from data capture to care giver use could have a latency of as much as a day, inclusive of

processing. For life-critical operations this latency could need to be as low as seconds.

   b) **Expansion**: Security technology such as encryption and error correcting codes often expand the size of the representation of data. This expansion should be minimized.
   c) **Wide Geographic Area**: The capability should operate with participants over a wide geographic area, ideally trans-continental if not inter-continental without an unacceptable degradation in latency.
   d) **Adaptability**: The architecture should be easy to adapt to multiple data formats, data networks and data store frameworks.
   e) **Extensibility**: The architecture should be able to be extended as new capabilities are needed or become practical, such as new methods to support encrypted computing. Also, the technology should be able to support different usage patterns, possibly with data from other sources, such as from digital health records.

3) **Cost**
   a) **Resource Efficient**: The computational requirements required to support the architecture should be as modest as possible, with little overhead due to the use of encrypted processing.
   b) **Scalability**: The capability should be able to support multiple users and participants and scale horizontally to support all the users of enterprise sized organizations.
   c) **Portable**: The should be easily ported to multiple commodity client and server types that are in broad current use.
   d) **Usability**: The capability should be easy to deploy, manage an update.

III. Secure Information Architecture

We use end-to-end lattice encryption to form the basis of our approach to realize our information architecture that addresses the above measures to encrypt data at the sensor, transmit the data to a cloud computing environment where processing is done on the data while it is still encrypted, and the encrypted results shared with intended recipients, without ever decrypted the data or sharing decryption keys. Current information architectures do not provide end-to-end encryption capabilities, thus creating vulnerabilities such as to insider attacks and creating an inability to lower costs by out-sourcing computing to commodity cloud computing environments.

Lattice encryption is a relatively new family of encryption technologies [5]. A key feature of lattice encryption technologies is that their security is based on the hardness of variants of the ”Shortest Vector Problem” [6]. As a result, lattice encryption schemes are generally considered post-quantum and are secure against attacks even from adversaries with practical quantum computing devices, in addition to adversaries with classical computing devices [7].

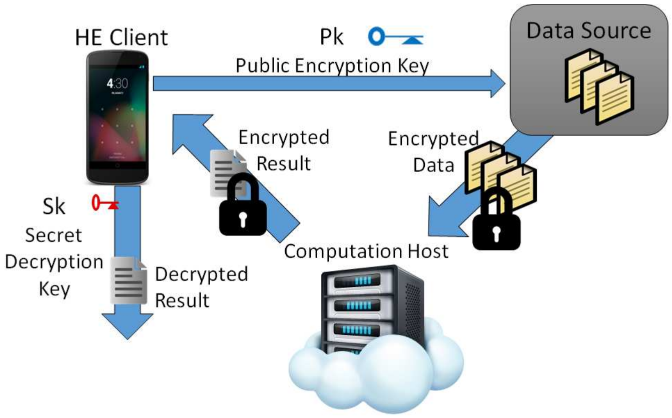

Fig. 2. Secure Computation Outsourcing

Recent results have shown lattice encryption schemes which have Homomorphic Encryption (HE) properties that make it theoretically possible to securely perform arbitrary computations on encrypted data without sharing keys decrypting the data [8]. To show the potential of this capability, consider the HE concept of operations model in Figure 2 where public encryption keys are given to a data source (such as a sensor) who encrypts data. Only encrypted data is given to a computation host without access to any decryption keys. The computation host runs computations on the encrypted data. The output of the host’s encrypted computation is shared with a client who has a decryption key. When the client uses the decryption key to decrypt the encrypted output, the resulting decrypted data is the same result as if the computation run by the host had been run on the original data without decryption.

Beyond recent encouraging HE results, we need to adapt the existing schemes to support our information architecture. Although recently discovered HE schemes can support arbitrary computations on encrypted data, many computations are impractically slow [9], but recent very positive results indicate HE designs [10] and implementations [11], [12], [13] are becoming increasingly practical. In fact, recent results show the practicality of classes of computations relevant to the processing that would need to be performed on wearable medical device data [14], [15].

Despite the practicality of HE, recent HE schemes do not have a native capability to delegate decryption other than by using the highly unpreferred method of sharing decryption keys. This changed recently with the demonstration of a lattice-based Proxy Re-Encryption (PRE) [16] which provide an approach to delegate decryption ability. Based on these results, we show in Subsection III-A applying HE to practically support important subsets computations encrypted wearable medical sensor data. In Subsection III-B we discuss modifying and integrating HE and PRE to enable our end-to-end encrypted architecture vision.

A. Secure Data Aggregation and Processing

Of particular interest to our architecture are variants of the NTRU lattice encryption scheme [17] which was not originally design for homomorphic encryption properties. NTRU spawned a family of related capabilities. See for example the LTV scheme [10] which demonstrates a very general and adaptable encrypted computing capability and provides a stronger security model which aligns with our information architecture vision. Early results on implementing a variant of the LTV scheme with a more traditional FHE computation model are shown in [13]. We build off the scheme in [13] to discuss the applicability of these approaches to wearable medical device processing.

We represent plaintext and ciphertext in our architecture as arrays of unsigned integers. For n a power of 2, we define the ring $R = \mathbb{Z}[x]/(x^n+1)$ (i.e., integer polynomials modulo x^n+1) where, and for any positive integer q, define the ciphertext space $R_q = R/qR$ (i.e., integer polynomials modulo x^n+1, with mod-q coefficients). The plaintext space is R_p for some integer $p \geq 2$. Concretely, we represent plaintext as a length-n vector of integers modulus p. Typically p is much smaller than 2^{64} and we typically choose p to be between 2 and 2^{10}. This ciphertext c is an n-element vector of mod-q integers. We concretely represent any ciphertext c in its evaluation, CRT or double-CRT representation [18]. For the sake of simplicity, power conservation and bandwidth limitations, we encrypt c at the sensors in its CRT representation where c is an n-element vector of mod-q integers. Typically n is set to 1024 and q is represented with 32 for initial security as discussed in [13].

The basis of the key generation and encryption operations in our scheme are as follows:

- KeyGen: choose a "short" $f \in R$ such that $f = 1 \bmod p$ and f is invertible modulo q. The "short" elements f is chosen from discrete Gaussian distributions. We output the secret key $sk = f$. Similarly, we choose a "short" $g \in R$ from a discrete Gaussian distribution. We output the public key $pk = h = g \cdot f^{-1} \bmod q$.
- $\mathsf{Enc}(pk = h, \mu \in R_p)$: choose a "short" $r \in R$ and a "short" $m \in R$ such that $m = \mu \bmod p$. The vectors r and m' are sampled from discrete Gaussian distributions and m can be chosen as $m = p \cdot m' + \mu$. We output $c = p \cdot r \cdot h + m \bmod q$.

The formulation of the key generation and encryption operations are key for generalizing the scheme to incorporate a PRE capability as in [16].

An important feature of the scheme is that enables the efficient aggregation of encrypted data at the smart phone access point to save on bandwidth. Each plaintext has n mod p "entries" to host data in each ciphertext c. We could organize the encoding of data into plaintext so that each sensor is assigned a different "entry" in each ciphertext. That is, if a heart rate monitor generates data every minute about the average heart rate hr, this data could be placed in the first plaintext entry as $[hr, 0, 0, \ldots, 0]$ and the array encrypted as c_1. If a prosthetic device generates failure alert data with a signal pa, this data could be placed in the second plaintext entry as $[0, pa, 0, \ldots, 0]$ and encrypted as c_2. We continue these assignments as needed to cover all wearable devices.

Because we are using a public key encryption scheme, the smart phone access point could be assigned a public key by the patient or even the patient's primary care provider. The smart phone could assign this public encryption key to the wearable medical devices to use for encrypting data so that the additive homomorphism would work. Because all of the data is encrypted with the same encryption key, data could be aggregated at the smart phone access point in its encrypted state using an additive homomorphism of the homomorphic encryption scheme. In this manner, each cellular access point could aggregate data $\{c_1, c_2, \ldots, c_n\}$ into $c_{agg} = c_1 + c_2 + \cdots + c_n \mod q$. Hence, we leverage the additive homomorphic properties of the encryption scheme to greatly reduce bandwidth consumption and have the access point regularly transmit c_{agg} to the computation host. Reported results in [13], [14] show that the encryption and ciphertext aggregation operations can be performed in milliseconds, leading to low latency due to these operations.

As aggregated ciphertexts $\{c^1_{agg}, c^2_{agg}, c^3_{agg}, \ldots\}$ are received by the computation host, we may not be able to practically perform *every* possible computation on the ciphertext datasets, but it is possible to perform a wide array of important operations such as linear regression and covariance computations [15]. These operations have a long history of use in basic statistical analyses which includes [19] as examples. Reported results in [15] indicate that the linear regression and covariance computations can be performed in the order of minutes depending on the size of the dataset. This experimental latency is non-trivial, indicating that the secure computations approach might not be reasonable for time-critical applications. However, statistical operations such as linear regression and covariance computations are usually only needed on large data sets which may need to be collected over long periods of time, indicating that the natural use cases for these encrypted operations are exactly when time is not critical and waiting several minutes for the result of a computation is not an issue.

These linear regression and covariance computations are non-trivially deep in their circuit representation and the ciphertext as we specified them can not support encrypted operations deeper than the simple addition operations. As such, we would need to use a special, computationally expensive operation on the ciphertext called bootstrapping. A recent efficient bootstrapping design is discussed in [20] and early implementations of it are discussed in [13]. As such, bootstrapping operations could be performed on the aggregated ciphertexts so that we can perform linear regression and covariance computations on them.

Discussions of concrete proof-of-concept implementations and parameter for these HE building blocks are given in [13], [15] and security proofs of the scheme is discussed in [10]. We can also leverage further representation and noise management optimizations from [21] which are applicable to our design.

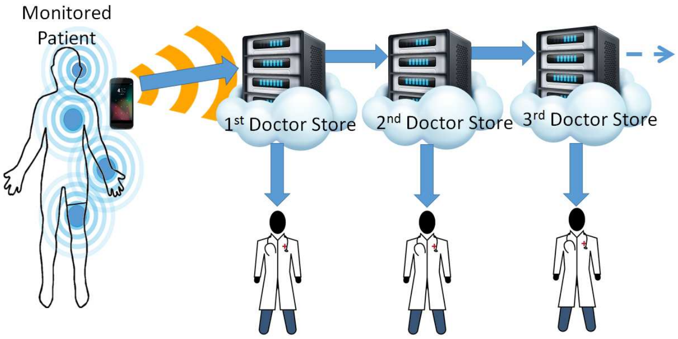

Fig. 3. Secure Access Redelegation

B. Secure Access Delegation

We can augment the public key HE encryption scheme we a capability to delegate data access with PRE. There are existing PRE designs based on lattice encryption [16]. The basic concept of operations for PRE as applied to a medical scenario is in Figure 3. The starting point for the use of the PRE scheme is the ciphertext resulting from the encrypted computations on an originating data stored as discussed in Subsection III-A. To illustrate the use of PRE, suppose the patient changes doctor and the new doctor needs access the patient's prior data. Rather than decrypt all of the patient's data at the first doctor's cloud provider and re-encrypting the data for the second doctor's provider, the second doctor could generate her own encryption/decryption key pair. The second doctor, using an out-of-band communication path, could send the new decryption key to the first doctor. There are existing security mechanisms to do this, such as using PGP public key encryption technologies. The first doctor then generates a "re-encryption" key from the two decryption keys. This re-encryption key is sent to the first doctor's cloud provider. The cloud provider, using the PRE ReEncryption operation, converts the encrypted data using the re-encryption key into a new encryption that could only be accessed with the second doctor's decryption key. None of the cloud providers could use the re-encryption key to decrypt any of the data, meaning that the first doctor is securely delegating decryption access to the second doctor without requiring the decryption (or bandwidth expensive downloading) of the patient data.

As can be seen in [16], the recent lattice-based PRE scheme uses a nearly identical key generation and encryption algorithm as the HE scheme in [10], [13], resulting in using the same decryption process:

- Dec$(sk = f, c \in R_q)$: compute $\bar{b} = f \cdot c \bmod q$, and lift it to the integer polynomial $b \in R$ with coefficients in $[-q/2, q/2)$. Output $\mu = b \bmod p$.

We parameterize applications of the cryptosystem so that the decryption operation is performed when there is a small ciphertext moduli. Experimental results from [16] indicate that the re-encryption process, using current designs and implementations runs in seconds, while [13] indicates that the decryption process runs in milliseconds. The prototype results indicate the basic feasibility of these approaches even for time-critical applications. The primary difference between the encryption and key generation processes between the HE and PRE schemes are the concrete parameter selection process. This indicates that the primary goal approach to concrete parameter selection would be one of guaranteeing that the relative ciphertext noise in the ciphertext is adequate for decryption at all times, even after encrypted computations are performed. This approach with parameter tradeoffs are discussed in [13].

An added approach of our integrated HE and PRE approach is that the PRE scheme is adequately generic and straightforward to generalize beyond wearable medical device data. For example, we could support the secure access delegation of encrypted unstructured data. Furthermore, the re-encryption process could conceivably be repeated indefinitely with the proper scheduling of the computationally expensive bootstrapping operation to manage the decryption noise in the ciphertext.

IV. Initial Experimentation

We performed an initial implementation of the lattice-based PRE cryptosystem in C++ using a hierarchical software architecture to enable rapid prototyping and simplify integration with embedded hardware. The design is modular and includes three major layers: (1) crypto, (2) lattice, and (3) arithmetic (primitive math). Encapsulation, low inter-module coupling, high intra-module cohesion, and maximum code reuse software engineering guidelines are followed when making any library changes. Lattice operations are decomposed into primitive arithmetic operations on integers, vectors, and matrices that are implemented in the primitive math layer. Along with basic modular operations, this layer includes efficient low-level modular mathemtic algorithms. The primitive math layer provides a high level of modularity allowing the library user to integrate with an existing low-level libraries or a custom hardware-specific implementation, such as an FPGA.

We ran initial experimentation on a commodity laptop and and obtained initial experimental runtime results that correspond to encrypting 1kb of data in 6.5ms, decrypt 1kb of data in 9.0ms and perform access delegation operations for 1kb of data in 52ms, all at a level of security with a work factor roughly corresponding to AES-128.

V. Related Work

There has been previous efforts to design network security architectures to reduce the risk of data leakage for health care organizations. Prime examples of this is [22], [23]. Unlike this prior work, a key feature of our proposed approach is the use of end-to-end encryption, thus greatly simplifying interactions between intermediate data hosts while also greatly reducing the risks of data leakage due to nefarious insiders and compromised devices. Thus, endpoint protection and endpoint encryption key management, as discussed in [22], [23] remain a key aspect of a security architecture because the endpoints are the only locations where data is accessible unencrypted.

However, we provide a more secure framework where data is protected by encryption at all locations and points in time.

Prior efforts such as [24] have similarly looked at issues of privacy-preserving data aggregation, but from an enterprise perspective where large organizations, such as hospitals, performed shared computations on sensitive data. Our architecture focuses on patient-level data security that support embedded mobile devices where bandwidth is an issue. However, the architecture we have designed can be generalized to support the private, enterprise-wide aggregation of patient level data, and then support the privacy-preserving inter-enterprise aggregation of that data. See for example [10] which provides a more general (but not experimentally evaluated) approach for data aggregation where participants perform a joint computation with results only accessible by common agreement.

VI. Conclusion

We presented a design for a secure information architecture which provides end-to-end encryption to protect data at all times. Experimental results have shown that our approach addresses many of the evaluation measures we present in Section II, and our approach focuses on privacy and confidentiality concerns. Our approach can be further augmented to provide important Integrity and Availability protections which form the other two legs of the CIA triad. For example, broad use of legacy cryptographic signing methods would bolster integrity and service replication would also bolster availability against denial-of-service attacks, for example.

However, a major benefit of our end-to-end encryption approach to provide confidentiality is that this scheme can be securely run on commodity cloud computing environments, and also greatly reduce the incidence of pernicious insider attacks. For example, even with encrypted computing hosted on proprietary servers, the encryption technologies limit the users who have access to the data to only the system administrators who have decryption key access. Taken together, this could greatly reduce the operational costs of highly regulated industries such as health-care where regulatory compliance restricts the ability to outsource computation to low cost cloud computing environments.

Looking forward, as HE becomes more practically capable, it is possible to see more general computations being practically supported in our architecture. For example, early results [21] shows the homomorphic evaluation of the AES circuits to "convert" AES encrypted data to HE representations without sharing keys in the clear. This would be a tremendous capability to increase the ability to integrate with broad legacy systems and result in additional end-to-end security capabilities.

References

[1] G. J. Annas, "Hipaa regulations-a new era of medical-record privacy?" *New England Journal of Medicine*, vol. 348, no. 15, pp. 1486–1490, 2003.

[2] R. Rauscher, "Cloud computing considerations for biomedical applications," *Healthcare Informatics, Imaging and Systems Biology, IEEE International Conference on*, vol. 0, p. 142, 2012.

[3] W. H. Maisel and T. Kohno, "Improving the security and privacy of implantable medical devices," *New England journal of medicine*, vol. 362, no. 13, p. 1164, 2010.

[4] J. Partala, N. Keränen, M. Särestöniemi, M. Hämäläinen, J. Iinatti, T. Jämsä, J. Reponen, and T. Seppänen, "Security threats against the transmission chain of a medical health monitoring system," in *e-Health Networking, Applications & Services (Healthcom), 2013 IEEE 15th International Conference on*, 2013, pp. 243–248.

[5] O. Goldreich, S. Goldwasser, and S. Halevi, "Public-key cryptosystems from lattice reduction problems," in *Advances in Cryptology-CRYPTO'97*. Springer, 1997, pp. 112–131.

[6] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM, 2009, pp. 333–342.

[7] D. Micciancio, "Lattice-based cryptography," in *Encyclopedia of Cryptography and Security*. Springer, 2011, pp. 713–715.

[8] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st annual ACM symposium on Theory of computing*, ser. STOC '09. New York, NY, USA: ACM, 2009, pp. 169–178. [Online]. Available: http://doi.acm.org/10.1145/1536414.1536440

[9] C. Gentry and S. Halevi, "Implementing Gentry's fully homomorphic encryption scheme," in *Advances in Cryptology EUROCRYPT 2011*, ser. Lecture Notes in Computer Science, K. Paterson, Ed. Springer Berlin / Heidelberg, 2011, vol. 6632, pp. 129–148.

[10] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM, 2012, pp. 1219–1234.

[11] Y. Doröz, Y. Hu, and B. Sunar, "Homomorphic aes evaluation using ntru." *IACR Cryptology ePrint Archive*, vol. 2014, p. 39, 2014.

[12] C. Gentry and S. Halevi, "HElib," https://github.com/shaih/HElib, 2014.

[13] K. Rohloff and D. B. Cousins, "A scalable implementation of fully homomorphic encryption built on NTRU," in *Proceedings of the 2nd Workshop on Applied Homomorphic Cryptography (WAHC)*, 2014.

[14] D. W. Archer and K. Rohloff, "Computing with data privacy: Steps toward realization," *IEEE Security & Privacy*, no. 1, pp. 22–29, 2015.

[15] D. Wu and J. Haven, "Using homomorphic encryption for large scale statistical analysis," 2012.

[16] D. Nuñez, I. Agudo, and J. Lopez, "Ntrureencrypt: An efficient proxy re-encryption scheme based on ntru," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*. ACM, 2015, pp. 179–189.

[17] J. Hoffstein, J. Pipher, and J. H. Silverman, "Ntru: A ring-based public key cryptosystem," in *Algorithmic number theory*. Springer, 1998, pp. 267–288.

[18] C. Gentry, S. Halevi, and N. Smart, "Homomorphic evaluation of the AES circuit," in *Advances in Cryptology CRYPTO 2012*, ser. Lecture Notes in Computer Science, R. Safavi-Naini and R. Canetti, Eds. Springer Berlin / Heidelberg, 2012, vol. 7417, pp. 850–867.

[19] Y. Salant, I. Gath, and O. Henriksen, "Prediction of epileptic seizures from two-channel eeg," *Medical and Biological Engineering and Computing*, vol. 36, no. 5, pp. 549–556, 1998.

[20] J. Alperin-Sheriff and C. Peikert, "Practical bootstrapping in quasilinear time," in *Advances in Cryptology–CRYPTO 2013*. Springer, 2013, pp. 1–20.

[21] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the aes circuit," in *Advances in Cryptology–CRYPTO 2012*. Springer, 2012, pp. 850–867.

[22] V. Gupta, M. Wurm, Y. Zhu, M. Millard, S. Fung, N. Gura, H. Eberle, and S. C. Shantz, "Sizzle: A standards-based end-to-end security architecture for the embedded internet," *Pervasive and Mobile Computing*, vol. 1, no. 4, pp. 425–445, 2005.

[23] R. Rauscher and R. Acharya, "A network security architecture to reduce the risk of data leakage for health care organizations," in *e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on*. IEEE, 2014, pp. 231–236.

[24] A. Andersen, K. Y. Yigzaw, and R. Karlsen, "Privacy preserving health data processing," in *e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on*. IEEE, 2014, pp. 225–230.