The Potential for Machine Learning Analysis over Encrypted Data in Cloud-based Clinical Decision Support – Background and Review

Jim Basilakis, Bahman Javadi, Anthony Maeder
School of Computer, Engineering and Mathematics
University of Western Sydney
Locked Bag 1797, Penrith 2751, New South Wales

j.basilakis@uws.edu.au

Abstract

In an effort to reduce the risk of sensitive data exposure in untrusted networks such as the public cloud, increasing attention has recently been given to encryption schemes that allow specific computations to occur on encrypted data, without the need for decryption. This relies on the fact that some encryption algorithms display the property of homomorphism, which allows them to manipulate data in a meaningful way while still in encrypted form. Such a framework would find particular relevance in Clinical Decision Support (CDS) applications deployed in the public cloud. CDS applications have an important computational and analytical role over confidential healthcare information with the aim of supporting decision-making in clinical practice. This review paper examines the history and current status of homomoprhic encryption and its potential for preserving the privacy of patient data underpinning cloud-based CDS applications.

Keywords: homomorphic encryption, clinical decision support, cloud computing.

1 Introduction

In Australia and worldwide, there is a growing impetus for adoption of electronic health records (EHRs) and personal health records (PHRs). The availability of this type of information in clinical care will act as a major driver for much needed IT reform in this sector, as well as opening up major opportunities for Clinical Decision Support (CDS) (AHIC Electronic Decision Support Systems Report 2009). The Personally Controlled Healthcare Record (PCEHR) is an example of a recent Australian initiative to promote patient information sharing across caregivers and healthcare provider institutions nationally (Pearce and Haikerwal 2010).

Against the backdrop of these proposed large scale health IT infrastructure changes, there has been a rapid uptake of cloud computing services by organisations that want to flexibly outsource their computational requirements according to individual demand (Buyya, Yeo et al. 2009).

Privacy and security concerns however dominate public cloud use in healthcare and constructing a CDS system to operate within this environment has considerable challenges both technically and socio-politically (Creating HIPAA-Compliant Medical Data Applications With AWS 2012, Schweitzer 2012, Demirkan and Delen 2013). As a result, attempts at outsourcing computing or analytical processing to third parties or using these services as repositories for data storage are significantly hindered. In fact, the recommendation is to either avoid the public cloud model altogether for this type of information or avoid exposing unencrypted data to cloud providers (Pearson and Benameur 2010, Puttaswamy, Kruegel et al. 2011). If public cloud resources are to be utilised under these circumstances, there is little choice other than to consider encrypting all sensitive data made available on the public cloud.

Standard encryption techniques typically prevent further interpretation or manipulation of data, requiring the ciphertext to be first downloaded and decrypted before computational analysis could be performed on the plaintext. Researchers instead have been looking towards applying *homomorphic encryption* (HE) methods as a solution for overcoming some of the privacy and data control issues in the cloud. Since being first demonstrated in 2009 by Craig Gentry (Gentry 2009), *fully HE* schemes that support both addition and multiplication operations (allowing arbitrary operations on encrypted data), are just becoming efficient enough to be considered practically useful. They are still predominantly limited to certain specific computations, for instance, finding the statistical mean over sets of encrypted quantitative data, searchable encryption and private information retrieval (Chow, Golle et al. 2009). Although arbitrary computation over encrypted data is far from a reality, there may be some middle ground where HE schemes with reduced computational capacity can still be exploited to facilitate secure computation over untrusted IT networks.

This paper introduces some background concepts surrounding CDS, cloud computing and information security. It then explores in greater detail the history and current literature on HE with a view to evaluating the practicality of enabling machine learning (ML) algorithms to operate over encrypted data in the cloud. Such a ML framework would satisfy the real requirement for performing complex and distributed CDS processing within the healthcare domain, while maintaining a high level of confidentiality.

2 Clinical Decision Support and the Cloud

CDS systems refer to any application that supports clinical decision-making and "provides clinicians or patients with clinical knowledge and patient-related information, intelligently filtered or presented at appropriate times to enhance patient care" (Osheroff, Teich et al. 2007). 'Support' in this case suggests aiding rather than making decisions. The general aim of CDS is to make data easier to assess, to foster optimal problem-solving by the clinician, or assist in the automation of manual processes (Greenes 2011). These aims have implications for improved patient safety and quality of care as well as improved efficiency and cost reductions for healthcare. ML techniques (such as Naïve Bayesian, Decision Tree and Neural Network Classifiers) are used in non-knowledge based CDS systems that make inferences from data patterns and do not rely on a human expert to input knowledge into the system directly (Berner 2007).

Since the introduction of CDS systems in hospital care in the early 1970s (Teije, Miksch et al. 2008), there has been slow progress to date towards adopting CDS systems into mainstream clinical care beyond simple reference information display and basic alerting systems (Beilby, Duszynski et al. 2005). This is generally indicative of the slow growth of IT in the healthcare sector, which remains dominated by paper-based information systems, where CDS systems have limited impact (AHIC Electronic Decision Support Systems Report 2009). With the emergence of national EHR and PHR systems, there will be a growing requirement for applying increasingly sophisticated analytical and health information management tools to support clinical decision-making and process improvement in healthcare practice (AHIC Electronic Decision Support Systems Report 2009). This trend is being actively enforced in the US through the American Recovery and Reinvestment Act's 'meaningful use' criteria. As part of criteria, CDS rule-based interventions operating on EHRs are systematically being mandated (Centers for Medicare & Medicaid Services 2014). Furthermore, the Department of Veterans Affairs in US has called on the industry to develop standardised interface specifications for CDS "functionality as a service" to be accessed by its integrated EHR (Service Interface Specifications for EHR Services: Federal Business Opportunities 2012).

Adoption of cloud computing in healthcare would allow relevant applications, such as the aforementioned proposed CDS functionality as a service, to have a much needed broadening of their processing and analytical capabilities applied across a wider range of shareable healthcare resources (Armbrust, Fox et al. 2010). Cloud computing is viewed as a "style of computing in which dynamically scalable and often virtualised resources are provided as a service over the Internet" (Dhar 2012). Cloud services are designed to flexibly respond to changing business requirements and represent a fundamental change in the way consumers and organisations utilise computing resources. The transition is away from owning the system to one where IT systems are accessed as a service when required (Soman 2011). Cloud computing is increasingly attractive to business entities that wish to take advantage of cost sharing, pay-per-use and on demand provisioning of large scale computing resources. Additionally, the ease of use, platform-independence and decentralised nature of cloud computing allows services to be more sharable across entities and are more suitable for group collaboration (Soman 2011).

There are several examples of data-intensive analytics and processed-based applications running on cloud-based frameworks that are available for research and for general industry (Sun and Aida 2010, Fehling, Leymann et al. 2011, Liu, Charif et al. 2012, Demirkan and Delen 2013). Few vendors however are piloting CDS cloud computing solutions for use in mainstream clinical care. Notable examples include cloud-based real-time monitoring and support for rural and remote critical care units (McGregor 2011), and a shared CDS knowledge repository for managing cardiovascular diseases and diabetes in a community cloud (Dixon, Simonaitis et al. 2013). In the US, the iDASH platform attempts to "level the playing field" by providing tools for sharing clinical and biological data in a privacy-preserving manner amongst the research community. Funded by the National Institutes of Health, the platform includes a high-performance computing environment enabled through a private Health Insurance Portability and Accountability Act (HIPAA)-certified cloud (Ohno-Machado, Bafna et al. 2012). Security appears to be achieved through anonymisation techniques, enterprise-grade application management, and project segregation leveraging virtualisation. Access to the platform is limited to iDASH centres or exported to other centres.

3 Cloud Security

As with any services containing sensitive health-related information, cloud-based systems are required to follow legislated provisions enforcing security protections surrounding access to patient and healthcare provider data. There are severe penalties in cost, patient safety, and provider reputation should any malicious or unintended security breaches occur.

A *private cloud* is able to securely isolate computer systems to within a single organisation's private network and away from unsolicited public access, while still retaining the benefits of an abstracted IT infrastructure offered by a cloud-based architecture (Zissis and Lekkas 2012). The private cloud however may not satisfy the processing power and economies of scale afforded to consumers using *public cloud* services due to the relatively lower numbers of computing and database resources that are typically available within a private network. Confining computing resources within a private cloud environment may also limit information sharing across a variety of healthcare institutions.

Even within private IT networks, attacks can occur as a result of 'insider threats', which are an often-underestimated risk to an organisation's information security. These threats include accidental disclosures, insider curiosity and data breach by insider (Rindfleisch 1997, Theoharidou, Kokolakis et al. 2005, Appari and Johnson 2010). The proportion of insider attacks compared to all healthcare provider privacy breaches recorded between 2005 to 2013 in the US was

approximately 17% (n= 165) (Privacy Rights Clearinghouse; Chronology of Data Breaches 2013). Insider threats could be envisioned for third-party cloud vendors entrusted with outsourced provider data.

In the public cloud however, the security concerns are considerable and rely on the fact that computing resources are exposed within a dynamic, distributed and shared environment, with consumers having little control in how data is accessed, proliferated or destroyed. Often there is a lack of transparency and accountability when cases of privacy breaches occur (Pearson and Benameur 2010). These properties of the public cloud paradigm complicate regulatory, governance and jurisdictional directives, which are very prominent in healthcare. Additionally, there are a large variety of opportunities available for accidental or intentional leakage of personal information from vulnerable IT systems accessed from publically sharable computing infrastructures. As virtualisation enables hardware resources to be shared across different users, this introduces new system vulnerabilities such as cross-virtual machine (VM) side-channel attacks that can result in extraction of secret keys and other confidential information from cloud instances by adversaries (Ristenpart, Tromer et al. 2009). Protections should be in place for these types of cloud-specific attacks.

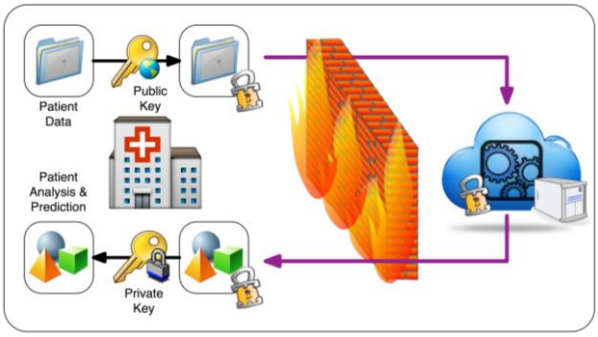

Figure 1: Scenario for Machine Learning Analysis over Encrypted Patient Data in the Cloud

HE is seen as an important technology for overcoming some of the privacy and data control issues when outsourcing computing and analytical processing to the public cloud. If ML analysis could be performed entirely on encrypted data, a cloud-based CDS application could potentially ensure the privacy of healthcare data that is used in both training and classification phases of the ML algorithm. Patient medical profiles from a hospital can be encrypted using the hospital's public key and sent to a third-party cloud vendor for ML analysis within a high-performance computing environment. The analysed result would be returned while still in its encrypted form and could only be revealed using the hospital's private key. The scenario is depicted in Figure 1. Such an approach may give both the healthcare institution as well as the public cloud vendor the confidence required for permitting processing of sensitive information outside of the hospital environment. Such an approach could also be used to strengthen HIPAA compliance or accreditation in a private cloud. In another scenario, a statistical model generated from current patient medical profiles from one hospital can be used to predict patient outcomes from another hospital based on their medical profiles. Ideally this outcome would be achieved using a protocol where neither party leaks any confidential patient details to the other party (Bost, Popa et al. 2014).

4 Homomorphic Encryption

4.1 Background

4.1.1 Definitions

HE schemes allow meaningful manipulations on encrypted data without knowing the secret key (Gentry 2010). HE schemes that allow simple operations on encrypted data have been known for some time. One of the first algorithms proposed by Rivest, Adleman et al. (1978) as having multiplicative homomorphic properties was the well-known Rivest, Shamir and Adleman (RSA) public-key encryption algorithm. The mathematical properties of the algorithm are such that the public-key encrypted form of two integers $m_1$and m_2, denoted as $E_{pk}(m_1)$ and $E_{pk}(m_2)$, when multiplied together would result in the encrypted form of the product of the two integers, namely $E_{pk}(m_1.m_2)$. Decrypting this product can reveal the correct solution to the actual (plaintext) product of two integers. The same public key is used in encryption of the two integers while the corresponding private key is used in decryption of the ciphertext product.

The ideal requirement is that a HE scheme is semantically secure. The notion of *semantic security* was introduced by Goldwasser and Micali (1982), which in general terms relates to the fact that an adversary should not be able to discover any partial information from a ciphertext. Semantic security is equivalent to the concept of *computational indistinguishability*, which is simpler to work with in formal proofs (Katz and Lindell 2007). Informally it represents a hypothetical situation where an adversary supplies two plaintexts, one of which is then randomly chosen, encrypted, and then handed back to the adversary to determine which of the two plaintexts was chosen. With the adversary's computational powers limited to running in *polynomial time*, the probability of choosing the correct plaintext should not be better than ½ (plus a negligible factor) (Fontaine and Galand 2009).

Since RSA is deterministic in its original form, it is not semantically secure. Any attempt to make it probabilistic breaks its homomorphic properties (Fontaine and Galand 2009). Additionally, it leaks some information including the fact that integers 0 and 1 have the same value in both plaintext and ciphertext form.

4.1.2 Partial Homomorphism

The first semantically secure HE scheme was described by Goldwasser and Micali (1984) and since then a number of additively and multiplicatively HE schemes have been described (Fontaine and Galand 2009, Vaikuntanathan 2011). Of particular note are the efficient and semantically secure HE schemes by Paillier (1999) and ElGamal (1985) (including their variants) which are additive and multiplicative respectively. These schemes display *partial homomorphism* because they do not

support both operations at the same time. A list of the most common schemes is presented in Table 1 from (Bailey, Bush et al. 2012).

Cryptosystem	Homomorphic Operation
RSA	Multiplication mod n
ElGamal	Multiplication, Exponentiation (by constant only)
Paillier	Addition, Subtraction, Multiplication (by constant only)
Glowasser-Micali	XOR
Benaloh	Addition, Subtraction
Naccache-Stern	Addition, Subtraction, Multiplication (by constant only)
Boneh-Goh-Nissim	Unlimited additions, one Multiplication

Table 1: List of Common Partial HE Schemes

4.1.3 Full Homomorphism

It was only until 2009 that a *fully homomorphic encryption* (FHE) scheme, which enables arbitrary computation over encrypted data, was demonstrated by Craig Gentry using *ideal lattices* in his PhD thesis (Gentry 2009). In general, these schemes are based on the intractable mathematical problems associated with *lattices*. The lattice points outline 'tilings' of the space of real numbers in n dimensions ($\mathbb{R}^n$). A lattice is constructed by combining n linearly independent vectors which forms the *basis* of the lattice, denoted by $V = \langle v_1, \cdots v_n \rangle$. Integral coefficients associated with the basis are used to span the entire lattice L (Goldreich, Goldwasser et al. 1996). That is:

$$L(V) = \left\{ \sum_i a_i v_i : a_i \in \mathbb{Z} \ \ for\ all\ i \right\}$$

Two of the most common (conjectured) computational problems in lattices include the Closest Vector Problem (CVP) and Shortest Vector Problem (SVP). In CVP, the problem relates to finding the lattice vector closest to a given target vector not on the lattice, while SVP is concerned with finding the shortest (non-zero) vector in the lattice (Goldreich, Micciancio et al. 1999). Both problems are known to be NP-hard to solve and grow proportionally (at least) to the exponent of the dimension of the lattice (Becker, Gama et al. 2013). In forming a public encryption scheme, a one-way computation function is required to be constructed, which for the CVP could mean adding a small error vector to a point in the lattice (the lattice point would have been initially mapped to a message). The advantage of lattices over the RSA and ElGamal encryption schemes are their reduced computation time for encryption and decryption, which are of the order of $O(k^2)$ for some security parameter k, since they are based only on simple polynomial multiplication. In comparison, the time complexity of RSA and ElGamal systems are of the order of $O(k^3)$. The disadvantage of lattice-based encryption schemes is in their public key size, which is $O(k^2)$ compared to $O(k)$ (Goldreich, Goldwasser et al. 1997).

Ideal lattices of the type employed by Craig Gentry in the development of FHEs are lattices with some additional algebraic structure, such as cyclic rotation of the vector set within the lattice basis. This allows for a more succinct representation of the n-dimensional lattice (using 1 vector) and can be processed more efficiently (Lyubashevsky 2008). Homomorphic operations are performed by the addition or multiplication of lattice points. As there is noise associated with generation of the ciphertext, the noise is roughly doubled and squared with the addition and multiplication operations respectively. This limits the amount of operations that can occur before successful decryption is no longer possible, resulting in a Somewhat Homomorphic Encryption (SWHE) Scheme (Kocabaş and Soyata).

The truly revolutionary idea of Gentry was to convert the SWHE scheme into a FHE Scheme using a *bootstrapping* method that repeatedly decrypts the ciphertext in a self-referential way (ie. recrypt) as a means of resetting the ciphertext noise. Each recryption operation step also allows only one arithmetic operation to be performed before the noise becomes too large and a recryption step is required before the next operation. A *squashing* process is necessary to transform the decryption scheme to one that is homomorphically equivalent but is simplified to allow bootstrapping. The repeated cycle of bootstrapping with an additional single arithmetic operation allows for the computation of arbitrary functions indefinitely (Gentry 2010). By handling all possible arithmetic functions, this scheme satisfies the criteria of homomorphism, although there is an additional requirement of *compactness* for it to be considered FHE. This means that the size of the cyphertext (and the time needed to decrypt it) does not grow with the complexity of the function being evaluated, but rather is dependent (polynomial) on the security parameter (Gentry 2010).

Even though Gentry's scheme demonstrated that FHE was possible, it was too inefficient to be considered practical both in terms of computation and storage. Estimates showed an 800,000 times storage expansion ratio was required for encrypting just one bit, and 99.9% of the total execution time was spent on the recryption operation (Kocabas, Soyata et al. 2013).

4.1.4 New generation HE schemes

There have since been a number of recent improvements to the original Gentry scheme making it more efficient and practical to use. In 2011, Gentry and Halevi removed the requirement for the squashing process but FHE was still based on ideal lattices. Concurrently Brakerski and Vaikuntanathan also removed squashing by exploiting Gentry's scheme but based it on the much more simple and efficient *learning with error* (LWE) problem. This was shown to be equivalent to the hardness of solving the SVP problem on any lattices (Brakerski and Vaikuntanathan 2014).

The LWE problem introduced by Regev (2009) states that a polynomial number of 'noisy' random linear combinations of coefficients of a secret vector s of n dimensions (modulo q) is hard to solve for s. A shortened example directly from Regev (2010) illustrating this problem shows that the inputs could be:

$$14s_1 + 15s_2 + 5s_3 + 2s_4 \approx 8 (\mathrm{mod} 17)$$
$$13s_1 + 14s_2 + 14s_3 + 6s_4 \approx 16 (\mathrm{mod} 17) ...$$

… and so on where each equation has a small additive error of say ±1. The answer for s is purported to be hard to recover due to the presence of the errors (the answer s=[0,13,9,11]). A slight variation of this scheme, the ring-LWE problem imposes some structure to the linear equations making them more compactly represented (using smaller keys), more efficient to compute, and can still be shown to be hard to solve on ideal lattices in the worst case (Regev 2010).

The new generation schemes developed using (± ring) LWE by Brakerski and Vaikuntanathan (2014) with later refinements from Brakerski, Gentry et al. (2012) removed the requirement for squashing and bootstrapping. While the scheme could perform additions homomorphically within the bounds of noise, a re-linearisation technique was used that required a different secret key to perform each level of multiplication under a new encryption. In this way up to L levels of multiplications could be performed using a pre-determined chain of L different keys as additional input. This new scheme became known as *levelled* FHE (Togan M 2014). Brakerski, Gentry et al. (2012) also introduced a noise management technique called *modulus switching* that relied on switching a ciphertext to one with a smaller modulus. This resulted in a concomitant decrease in the magnitude of the noise without involving a secret key and allowed an exponential increase in the number of multiplications that could occur (n levels instead of log n) before bootstrapping would be required (Vaikuntanathan 2011). This particular scheme became known as the *BGV* implementation of levelled FHE after its authors.

A number of key optimisations and batch techniques have reduced overall computation complexity and increased efficiency of these FHE schemes. This included ciphertext packing techniques for combining multiple ciphertexts into a single ciphertext. Packing techniques were developed for ideal lattices, binary vectors and (± ring) LWE problems (Zhou and Wornell 2014). Single Instruction Multiple Operations (SIMD) proposed by Smart and Vercauteren (Smart and Vercauteren 2010) applied a variation of the ciphertext packing technique and were adopted by some of the newer schemes to achieve parallelisation of repeated operations by packing these bits into the same cyphertext . Various additional algorithm and permutation optimisations appear in specialised schemes to improve performance (Halevi and Shoup 2014).

Overall there is a complexity associated with lattice-based cryptography schemes and those based on the LWE problem. Setting up the correct secure encryption environment involves consideration of a number of inter-related parameters (Naehrig, Lauter et al. 2011). In addition, the schemes break down computational tasks into multiple binary operations making it difficult to practically accommodate computation and communication requirements (Zhou and Wornell 2014).

Van Dijk, Gentry et al. (2010) introduced the *DGHV* scheme which is a conceptually simpler as it relies only on simple integer operations rather than depending on single bit manipulations over plaintext. In this way, a number of standard computations could be supported. The scheme was based on the hardness of solving the *approximate Greatest Common Divisor (GCD) problem*. The assumption here is that it is normally simple to solve for the GCD of two integers using Euclid's theorem, but this is not the case when small errors are associated with each integer (Gentry 2010).

The initial DGHV scheme was still impractical as it involved the use of very large public keys. There have since been a number of improvements to the scheme including reduction of the public key size (Coron, Mandal et al. 2011) and batch processing of a vector of plaintext bits as a single ciphertext (Cheon, Coron et al. 2013). Cryptosystems that rely on the approximate GCD problem however appear to be less efficient compared to equivalent (security-wise) lattice-based schemes (Gentry 2010).

4.2 Towards Practical HE

4.2.1 Somewhat HE Schemes

The performance of FHE operations, have improved significantly from a few hours in 2010 to a few milliseconds by 2012 (Fau, Sirdey et al. 2013) which has given much hope to the ultimate practically of these schemes. A current area of intense research is the application of SWHE encryption schemes, which are more efficient and optimised compared to FHE schemes, at the expense of reduced (but sufficient) functionality (Naehrig, Lauter et al. 2011, Sen 2013). Even before the discovery of FHE, partial HE schemes have been exploited for their limited computational capabilities. Most HE schemes however (partial HE, SWHE and FHE), so far have been predominantly focused on specialised functions applied to well-known application areas. A list of only some of these areas, include: electronic voting/auctions/lotteries (Fouque, Poupard et al. 2001, Abe and Suzuki 2002), private set intersection (Kerschbaum 2012), private information retrieval (Sion and Carbunar 2007), data aggregation in wireless networks (Acharya, Girao et al. 2005, Westhoff, Girao et al. 2006), watermark and fingerprint schemes (Kuribayashi and Tanaka 2005).

4.2.2 Healthcare Applications

The healthcare setting is commonly considered a strong candidate for HE although many of the suggested schemes remain tightly coupled to a particular healthcare use case. In healthcare sector, typically the data rather than the computations are considered sensitive or private, compared to other industries such as the financial and advertising sectors (Naehrig, Lauter et al. 2011). Currently the implementation status of most health-based HE schemes has been at a conceptual level only or there may be existing prototypes for performance demonstration purposes. Many earlier systems are predominantly based on the additively homomorphic and semantically secure Pallier cryptosystem. Katzenbeisser and Petkovic (2008) for example, describe using the scheme to securely calculate an inner product between an encrypted and plaintext vector to determine the degree of correlation between them. This function could be used to privately match a particular patient disease profile (mapped into some binary form) against either a reference knowledge set of diseases, or specialist disease expertise or other patients with similar disease profiles for a

community (P2P) networking application. Pallier-based HE schemes have also been described in the context of privacy preserving statistical analysis of ubiquitous (wearable sensor) health data (Drosatos and Efraimidis 2011); ECG signal classification using private linear branching programs (as a generalisation of decision trees) (Barni, Failla et al. 2009) and neural network techniques (Lagendijk, Erkin et al. 2013); as well as private genomic data mining for use in personalised medicine (Kantarcioglu, Jiang et al. 2008).

More recently, the application of levelled FHE schemes have been examined in the context of healthcare such as in the real-time privacy-preserving analysis of medical data acquisition devices over the cloud (Kocabaş and Soyata , Kocabas, Soyata et al. 2013). It has also been applied to privacy-preserving predictive analysis of medical data based on logistic regression and the Cox proportional hazard model running in the (Microsoft's Windows Azure) cloud (Bos, Lauter et al. 2014). Finally, FHE schemes have also been explored at a conceptual level only (and in a very unclear way) in relation to integration with EHRs (Soubhagya, Mini et al. 2013, Ikuomola and Arowolo 2014) including the PCEHR (Begum, Mamun et al. 2013).

4.2.3 Generalising HE Algorithms

A key factor providing FHE protocols with the capability to solve real work problems depends on the extent to which they can support general computing functions of practical interest over encrypted data through sufficiently expressive composable primitives. When combined in various ways and without loss of generality, these primitives would allow for a much wider and more practical implementation scope. A number of areas have been examined in this space including secret program execution, database queries and ML algorithms.

In secret program execution, at the processor level, primitives have been contemplated that would allow arbitrary and dynamic program executions through the combination of memory access logic, arithmetic and encrypted branching operations (Brenner, Wiebelitz et al. 2011, Brenner, Perl et al. 2012). At an application level, primitives such as runtime data-dependent program control (if-then-else expressions) and integer manipulations provide for a more natural expression of high-level algorithms such as array summation, bubble-sort and Fast Fourier Transform calculations (Fau, Sirdey et al. 2013).

Supporting private database queries over encrypted data has received the most attention in this space due to the very large potential for the secure storage and access of encrypted data in the cloud without the requirement for decryption. The general types of operations that are required to be supported include complex selection, range, join, and aggregation operations, and FHE primitives have been shown to support these general database queries (Wang, Agrawal et al. 2012, Boneh, Gentry et al. 2013).

5 Privacy-Preserving ML Algorithms

In generalising machine learning and statistical algorithms, when using a SWHE or levelled HE scheme, one is limited to functions that can be expressed as a low degree polynomial (where there are many additions and a small number of multiplications). This includes simple statistical functions such as mean and standard deviation (Naehrig, Lauter et al. 2011) as well as non-trivial machine learning algorithms. This includes binary classifiers such as logistical regression, Linear Means (LM) classifier and Fisher's Linear Discriminant Classifier (Graepel, Lauter et al. 2013). Note that there is no efficient way to do divisions or square roots, which would otherwise require more expensive interactive protocols between the data owner and the cloud service provider of the outsourced algorithm. This would appear to preclude algorithms for which there are no division-free integer derivations. Suggestions for extending these methods to other machine learning algorithms were made using approximation or decomposition methods, but were not demonstrated (Graepel, Lauter et al. 2013). The performance of the LM classifier on 30 features from a public breast cancer dataset using 100 training and test vectors was achieved in approximately 6 seconds.

Bost, Popa et al. (2014) took quite a different approach to generating building blocks that would construct classifiers securely. Focusing on the classification rather than training phases, what made the authors' approach so unique is that they combine two partially homomorphic cryptosystems and a levelled FHE and make frequent use of interactive protocols in an effort to make the underlying primitives more efficient and flexible, thereby supporting a wider scope of secure machine learning algorithms. The authors also ensure that these protocols are provably semantically secure and in such a way that maintains the privacy of both the client's private input and the server's classifier model (privacy-preserving classification). The three encryption schemes used in this approach are the Paillier, BGV and the Goldwasser-Micali cryptosystems. While the first two are examples of partial (additive) and levelled fully homomorphic cryptosystems respectively, the homomorphic property of the Goldwasser-Micali cryptosystem is exclusive-or (XOR) addition modulo 2 (of an encryption of a bit), based on the Quadratic Residuosity problem (Goldwasser and Micali 1982). The machine-learning primitives that are supported using this approach include a comparison protocol to find the larger value of two encrypted inputs, and argmax protocol to find the index of the largest value from a list of encrypted integers, with the latter relying on the former protocol. Both protocols require back and forth interactions between a client and server. Two other supported primitives are a trivial computation of a private dot product, as well as a protocol for switching between encryption schemes. The latter protocol is also interactive, allowing primitives to be securely combined in a variety of ways to achieve the broad complement of machine learning algorithms.

The authors illustrate the primitives being used to build hyperplane decision classifiers (which covers perceptrons, least squares, Fisher linear discriminant and support vector machines), Naïve Bayes and Decision Trees. They also demonstrate using AdaBoost as a technique for combining the classifiers. Despite the use of interactive protocols in constructing the three classification groups, the use of relatively more efficient

partial HE schemes appears to make these algorithms more efficient overall. The use of levelled FHE is limited to enabling decision tree analysis. A number of medical datasets have been used to demonstrate system performance, achieving classification times of milliseconds to seconds, with a predominant amount of time being dedicated to communication during interactions.

6 Discussion

In review of the research, a number of techniques have been uncovered for potentially enabling the practical use of homomorphic cryptosystems in ML analysis that could be optimised for use in the public cloud environment. The field is rapidly evolving and there are many opportunities for combining primitives to enable different possible secure operations. An example here is the approach from (Zhou and Wornell 2014) who demonstrated how to process integers more efficiently for certain application areas such as feature extraction, recognition, classification and data aggregation. An interesting question posed by the authors also applies here: determining how to best divide machine learning computational tasks into "fundamental operations that minimize the overall communication and computation cost".

Technique	References
Automatic parameter selection in FHE for correctness and security against known lattice attacks.	(Naehrig, Lauter et al. 2011, Lepoint and Naehrig 2014)
Dealing with real numbers in arithmetic operations.	(Bost, Popa et al. 2014)
Use of Fast Fourier Transforms to speed up addition calculations.	(Bos, Lauter et al. 2014)
Message encoding techniques for ease of performing HE operations.	(Naehrig, Lauter et al. 2011)
Message encoding techniques for dealing with large integers.	(Wu and Haven 2012)
Batch techniques to improve computation performance.	(Wu and Haven 2012)

Table 2: Practical Optimisations Techniques for HE

A variety of practical considerations and optimisation techniques would be required of any ML system to be successfully applied in any real sense. Table 2 lists some of the techniques encountered including the references that mention using the technique.

Consideration should also be given to cloud optimisation factors, including high bandwidth requirements when uploading encrypted data over the Internet. The ciphertext size of an FHE scheme like BGV can be very large due to the large ciphertext expansion (thousands to millions), making transport over the network impractical. It has been suggested (Kocabaş and Soyata , Naehrig, Lauter et al. 2011, Kocabas, Soyata et al. 2013, Lepoint and Naehrig 2014) using a block cipher like the Advanced Encryption Standard (AES) to upload data to the cloud (with a minimal expansion ratio) and then evaluating the AES homomorphically to perform the decryption to FHE which has been achieved by Gentry, Halevi et al. (2012). This is a computationally intensive step taking around 36 hours (in one implementation) in total but could be reduced to 40 minutes per AES block using batching techniques. This step however is a one-off calculation and could be processed offline using less expensive cloud resources. It has been suggested by Kocabas, Soyata et al. (2013) that this method could also be used to balance storage vs computation requirements on the cloud by converting any portion of data dynamically from the permanently stored AES encrypted form to the (memory intensive) FHE form to perform the computation. When sending the computed result back to the client in encrypted form, a dimension reduction technique can be used convert the message into a shorter ciphertext that can no longer support any further homomorphisms. In this way the bandwidth issues encountered during upload and download of the data to the cloud can be avoided. Lepoint and Naehrig (2014) suggested the use of a more lightweight family of block ciphers, called SIMON (AlKhzaimi and Lauridsen 2013) that were more suitable to homomorphic evaluation and demonstrated significant improvement in transformation rates to a FHE scheme.

In HE, considerations of a more fundamental nature need to be factored into a scheme's practical implementation. In developing any cryptography protocol, formal proofs are absolutely essential for achieving claims about a particular level of security. Formal proofs are strongly recommended over using a hit-and-miss strategy, which all too often reveals later down the track vulnerabilities from subtle flaws in the encryption scheme. Typically, the consequent loss of confidentiality cannot be taken back (Lindell and Pinkas 2009).

When evaluating the security strength of cryptography systems, it is important to take into consideration the power of the adversary and the context of allowable behaviour. The highest level of security that can be attained in the setting of HE is one where the corrupt behaviour of the adversary does not deviate away from the encryption protocols and only has access to the information of all corrupt parties (Fontaine and Galand 2009). This behaviour model is known as *semi-honest (honest-but-curious* or *passive)*. Furthermore, a HE scheme could not offer protection against an adversary who has the ability to generate decryptions from ciphertexts of their choosing (as often as they wish) under the same scheme and given encryption key (Fontaine and Galand 2009). Overall, HE can only imply a security guarantee equivalent to the basic notion of semantic security for public key encryption.

Finally, HE schemes alone cannot enforce all the privacy requirements of a common cloud. An important limitation is that all arbitrary encrypted operations are restricted to within the domain of a single public key, thereby making it difficult to support different levels of data access control. In this setting alternative schemes would require consideration (Van Dijk and Juels 2010, Wang, Agrawal et al. 2012).

7 Conclusion

A privacy-preserving machine-learning framework based on HE is particularly relevant to the healthcare context. It would allay some of the security concerns that are a significant impediment to outsourcing of computing or

analytical processing of highly confidential patient information, to the public cloud. HE appears to be very a viable and rapidly evolving technology that has significant potential to enable the broadening of CDS processing and analytical capabilities across a wider spectrum of shareable healthcare resources over the cloud. Such CDS services would be more adaptable and responsive to changing business requirements as well as become more accessible to the general user while still preserving confidentiality.

8 References

Abe, M. and K. Suzuki (2002): M+ 1-st price auction using homomorphic encryption. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences* **86**(1): 136-141.

Acharya, M., J. Girao and D. Westhoff (2005): Secure comparison of encrypted data in wireless sensor networks. *Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 47-53, IEEE.

AHIC Electronic Decision Support Systems Report: Australian Health Information Council. http://newsletters.gpqld.com.au/content/Document/iNewsletters/iNewsletter 36 (11 Feb)/art2.pdf. Accessed 24 June 2014.

AlKhzaimi, H. and M. M. Lauridsen (2013): Cryptanalysis of the SIMON Family of Block Ciphers. *IACR Cryptology ePrint Archive* **2013**: 543.

Appari, A. and M. E. Johnson (2010): Information security and privacy in healthcare: current state of research. *International journal of Internet and enterprise management* **6**(4): 279-314.

Armbrust, M., A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia (2010): A view of cloud computing. *Communications of the ACM* **53**(4): 50-58.

Homomorphic Encryption (Poster): http://vanets.vuse.vanderbilt.edu/dokuwiki/lib/exe/fetch.php?media=teaching:poster12.10.pdf. Accessed 24 June, 2014.

Barni, M., P. Failla, V. Kolesnikov, R. Lazzeretti, A.-R. Sadeghi and T. Schneider (2009): Secure evaluation of private linear branching programs with medical applications. In *Computer Security–ESORICS 2009*. 424-439. Springer.

Becker, A., N. Gama and A. Joux (2013): Solving shortest and closest vector problems: The decomposition approach, Cryptology Eprint. Report 2013/685.

Begum, M., Q. Mamun and M. Kaosar (2013): A Privacy-Preserving Framework for Personally Controlled Electronic Health Record (PCEHR) System. *Proceedings of the 2nd Australian eHealth Informatics and Security Conference*, SRI Security Research Institute.

Beilby, J. J., A. J. Duszynski, A. Wilson and D. A. Turnbull (2005): Electronic Decision Support Systems At Point of Care: Trusting the Deus Ex Machina. *The Medical Journal of Australia* **183**(2): 99-100.

Berner, E. S. (2007): *Clinical Decision Support Systems*. Springer.

Boneh, D., C. Gentry, S. Halevi, F. Wang and D. J. Wu (2013): Private database queries using somewhat homomorphic encryption. In *Applied Cryptography and Network Security*. **7954:** 102-118. Springer.

Bos, J. W., K. Lauter and M. Naehrig (2014): Private Predictive Analysis on Encrypted Medical Data. *Journal of Biomedical Informatics* **50**: 234-243.

Bost, R., R. A. Popa, S. Tu and S. Goldwasser (2014): Machine Learning Classification over Encrypted Data. *Cryptology ePrint Archive, Report 2014/331*.

Brakerski, Z., C. Gentry and V. Vaikuntanathan (2012): (Leveled) fully homomorphic encryption without bootstrapping. *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 309-325, ACM.

Brakerski, Z. and V. Vaikuntanathan (2014): Efficient fully homomorphic encryption from (standard) LWE. *SIAM Journal on Computing* **43**(2): 831-871.

Brenner, M., H. Perl and M. Smith (2012): How practical is homomorphically encrypted program execution? an implementation and performance evaluation. *11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 375-382, IEEE.

Brenner, M., J. Wiebelitz, G. von Voigt and M. Smith (2011): Secret program execution in the cloud applying homomorphic encryption. *Proceedings of the 5th IEEE International Conference on Digital Ecosystems and Technologies Conference (DEST)*, 114-119, IEEE.

Buyya, R., C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic (2009): Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems* **25**(6): 599-616.

Centers for Medicare & Medicaid Services: http://www.cms.gov. Accessed 26 August, 2014.

Cheon, J. H., J.-S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi and A. Yun (2013): Batch fully homomorphic encryption over the integers. In *Advances in Cryptology–EUROCRYPT 2013*. 315-335. Springer.

Chow, R., P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka and J. Molina (2009): Controlling data in the cloud: outsourcing computation without outsourcing control. *Proceedings of the 2009 ACM workshop on Cloud computing security*, 85-90, ACM.

Coron, J.-S., A. Mandal, D. Naccache and M. Tibouchi (2011): Fully homomorphic encryption over the integers with shorter public keys. In *Advances in Cryptology–CRYPTO 2011*. 487-504. Springer.

Creating HIPAA-Compliant Medical Data Applications With AWS: http://awsmedia.s3.amazonaws.com/AWS_HIPAA_Whitepaper_Final.pdf. Accessed 24 June, 2014.

Demirkan, H. and D. Delen (2013): Leveraging the capabilities of service-oriented decision support

systems: Putting analytics and big data in cloud. *Decision Support Systems* **55**(1): 412-421.

Dhar, P. (2012): Cloud computing and its applications in the world of networking. *International Journal of Computer Science Issues (IJCSI)* **9**(1): 430-433.

Dixon, B. E., L. Simonaitis, H. S. Goldberg, M. D. Paterno, M. Schaeffer, T. Hongsermeier, A. Wright and B. Middleton (2013): A pilot study of distributed knowledge management and clinical decision support in the cloud. *Artificial intelligence in medicine* **59**(1): 45-53.

Drosatos, G. and P. S. Efraimidis (2011): Privacy-preserving statistical analysis on ubiquitous health data. In *Trust, Privacy and Security in Digital Business*. 24-36. Springer.

ElGamal, T. (1985): A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology*. **196:** 10-18. Springer.

Fau, S., R. Sirdey, C. Fontaine, C. Aguilar-Melchor and G. Gogniat (2013): Towards practical program execution over fully homomorphic encryption schemes. *Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, 284-290, IEEE.

Fehling, C., F. Leymann, D. Schumm, R. Konrad, R. Mietzner and M. Pauly (2011): Flexible process-based applications in hybrid clouds. *International Conference on Cloud Computing (CLOUD)*, 81-88, IEEE.

Fontaine, C. and F. Galand (2009): A survey of homomorphic encryption for nonspecialists. *Journal on Information Security* **1**: 41-50.

Fouque, P.-A., G. Poupard and J. Stern (2001): Sharing decryption in the context of voting or lotteries. In *Financial Cryptography*. **1962:** 90-104. Springer.

Gentry, C. (2009): A fully homomorphic encryption scheme. Ph. D. thesis. Stanford University, California.

Gentry, C. (2009): Fully homomorphic encryption using ideal lattices. *Proceedings of the 41st annual ACM symposium on Theory of computing (STOC)*, **9**:169-178, ACM.

Gentry, C. (2010): Computing arbitrary functions of encrypted data. *Communications of the ACM* **53**(3): 97-105

Gentry, C., S. Halevi and N. P. Smart (2012): Homomorphic evaluation of the AES circuit. In *Advances in Cryptology–CRYPTO 2012*. **7417:** 850-867. Springer.

Goldreich, O., S. Goldwasser and S. Halevi (1996): Collision-free hashing from lattice problems. *Electronic Colloquium on Computational Complexity (ECCC)*, **3**:236-241.

Goldreich, O., S. Goldwasser and S. Halevi (1997): Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology—CRYPTO'97*. 112-131. Springer.

Goldreich, O., D. Micciancio, S. Safra and J. P. Seifert (1999): Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters* **71**(2): 55-61.

Goldwasser, S. and S. Micali (1982): Probabilistic encryption & how to play mental poker keeping secret all partial information. *Proceedings of the 14th annual ACM symposium on Theory of computing*, 365-377, ACM.

Goldwasser, S. and S. Micali (1984): Probabilistic encryption. *Journal of computer and system sciences* **28**(2): 270-299.

Graepel, T., K. Lauter and M. Naehrig (2013): ML confidential: Machine learning on encrypted data. In *Information Security and Cryptology–ICISC 2012*. **7839:** 1-21. Springer.

Greenes, R. A. (2011): *Clinical decision support: the road ahead*. Academic Press.

Halevi, S. and V. Shoup (2014): Algorithms in HElib, Cryptology ePrint Archive, Report 2014/106.

Ikuomola, A. J. and O. O. Arowolo (2014): Securing Patient Privacy in E-Health Cloud Using Homomorphic Encryption and Access Control. *International Journal of Computer Networks and Communications Security* **2**(1): 15-21.

Kantarcioglu, M., W. Jiang, Y. Liu and B. Malin (2008): A cryptographic approach to securely share and query genomic sequences. *Information Technology in Biomedicine, IEEE Transactions on* **12**(5): 606-617.

Katz, J. and Y. Lindell (2007): *Introduction to modern cryptography: principles and protocols*. CRC Press.

Katzenbeisser, S. and M. Petkovic (2008): Privacy-preserving recommendation systems for consumer healthcare services. *3rd International Conference on Availability, Reliability and Security (ARES)*, 889-895, IEEE.

Kerschbaum, F. (2012): Outsourced private set intersection using homomorphic encryption. *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, 85-86, ACM.

Kocabaş, Ö. and T. Soyata Medical Data Analytics in the cloud using Homomorphic Encryption. In *Handbook of Research on Cloud Infrastructures for Big Data Analytics*. 471-488. IGI Global.

Kocabas, O., T. Soyata, J.-P. Couderc, M. Aktas, J. Xia and M. Huang (2013): Assessment of Cloud-based Health Monitoring using Homomorphic Encryption. *Proceedings of the 31st IEEE international conference on computer design*, 443-446, IEEE.

Kuribayashi, M. and H. Tanaka (2005): Fingerprinting protocol for images based on additive homomorphic property. *Image Processing, IEEE Transactions on* **14**(12): 2129-2139.

Lagendijk, R. L., Z. Erkin and M. Barni (2013): Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation. *Signal Processing Magazine, IEEE* **30**(1): 82-105.

Lepoint, T. and M. Naehrig (2014): A comparison of the homomorphic encryption schemes FV and YASHE. In *Progress in Cryptology–AFRICACRYPT 2014*. **8469:** 318-335. Springer.

Lindell, Y. and B. Pinkas (2009): Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality* **1**(1): 59-98.

Liu, H., Y. Charif, G. Jung, A. Quiroz, F. Goetz and N. Sharma (2012): Towards simplifying and automating business process lifecycle management in hybrid clouds. *19th International Conference on Web Services (ICWS)*, 592-599, IEEE.

Lyubashevsky, V. (2008): Lattice-based identification schemes secure under active attacks. In *Public Key Cryptography–PKC 2008*. **4939:** 162-179. Springer.

McGregor, C. (2011): A cloud computing framework for real-time rural and remote service of critical care. *24th International Symposium on Computer-Based Medical Systems (CBMS)*, 1-6, IEEE.

Naehrig, M., K. Lauter and V. Vaikuntanathan (2011): Can homomorphic encryption be practical? *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, 113-124, ACM.

Ohno-Machado, L., V. Bafna, A. A. Boxwala, B. E. Chapman, W. W. Chapman, K. Chaudhuri, M. E. Day, C. Farcas, N. D. Heintzman and X. Jiang (2012): iDASH: integrating data for analysis, anonymization, and sharing. *Journal of the American Medical Informatics Association* **19**: 196-201.

Osheroff, J. A., J. M. Teich, B. Middleton, E. B. Steen, A. Wright and D. E. Detmer (2007): A roadmap for national action on clinical decision support. *Journal of the American medical informatics association* **14**(2): 141-145.

Paillier, P. (1999): Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology—EUROCRYPT'99*. **1592:** 223-238. Springer.

Pearce, C. and M. C. Haikerwal (2010): E-health in Australia: time to plunge into the 21st century. *The Medical journal of Australia* **193**(7): 397.

Pearson, S. and A. Benameur (2010): Privacy, security and trust issues arising from cloud computing. *2nd International Conference on Cloud Computing Technology and Science (CloudCom)*, 693-702, IEEE.

Privacy Rights Clearinghouse; Chronology of Data Breaches: http://www.privacyrights.org/data-breach. Accessed Sep, 2013.

Puttaswamy, K. P. N., C. Kruegel and B. Y. Zhao (2011): Silverline: toward data confidentiality in storage-intensive cloud applications. *Proceedings of the 2nd ACM Symposium on Cloud Computing (SOCC)*, 1-13, ACM.

Regev, O. (2009): On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)* **56**(6): 1-40.

Regev, O. (2010): The Learning with Errors Problem (Invited Survey). *25th Annual Conference Computational Complexity (CCC)*, 191-204, IEEE.

Rindfleisch, T. C. (1997): Privacy, information technology, and health care. *Communications of the ACM* **40**(8): 92-100.

Ristenpart, T., E. Tromer, H. Shacham and S. Savage (2009): Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. *Proceedings of the 16th ACM Conference on Computer and Communications Security*, 199-212, ACM.

Rivest, R. L., L. Adleman and M. L. Dertouzos (1978): On data banks and privacy homomorphisms. *Foundations of secure computation* **4**(11): 169-180.

Schweitzer, E. J. (2012): Reconciliation of the cloud computing model with US federal electronic health record regulations. *Journal of the American Medical Informatics Association* **19**(2): 161-165.

Sen, J. (2013): *Homomorphic Encryption: Theory & Applications*. InTech.

Service Interface Specifications for EHR Services: Federal Business Opportunities: Federal Business Opportunities. https://http://www.fbo.gov/?s=opportunity&mode=form&tab=core&id=ef925a47c8027505721c1a4fdefab953&_cview=0. Accessed 26 August, 2014.

Sion, R. and B. Carbunar (2007): On the computational practicality of private information retrieval. *In Proceedings of the Network and Distributed Systems Security Symposium (NDSS)*.

Smart, N. P. and F. Vercauteren (2010): Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Public Key Cryptography–PKC 2010*. **6056:** 420-443. Springer.

Soman, A. K. (2011): *Cloud-based Solutions for Healthcare IT*. Science Publishers.

Soubhagya, B., V. G. Mini and J. A. Celin (2013): A Homomorphic Encryption Technique for Scalable and Secure Sharing of Personal Health Record in Cloud Computing. *International Journal of Computer Applications* **67**(11): 40-44.

Sun, H. and K. Aida (2010): A Hybrid and Secure Mechanism to Execute Parameter Survey Applications on Local and Public Cloud Resources. *2nd International Conference on Cloud Computing Technology and Science (CloudCom)*, 118-126, IEEE.

Teije, A. t., S. Miksch and P. Lucas (2008): *Computer-based medical guidelines and protocols: a primer and current trends*. IOS Press.

Theoharidou, M., S. Kokolakis, M. Karyda and E. Kiountouzis (2005): The insider threat to information systems and the effectiveness of ISO17799. *Computers & Security* **24**(6): 472-484.

Togan M, P. s. C. (2014): Comparison-Based Computations Over Fully Homomorphic Encrypted Data. *10th International Conference on Communications (COMM)*, 1-6, IEEE.

Vaikuntanathan, V. (2011): Computing blindfolded: New developments in fully homomorphic encryption. *52nd Annual Symposium on Foundations of Computer Science (FOCS)*, 5-16, IEEE.

Van Dijk, M., C. Gentry, S. Halevi and V. Vaikuntanathan (2010): Fully homomorphic encryption over the integers. In *Advances in Cryptology–EUROCRYPT 2010*. **6110:** 24-43. Springer.

Van Dijk, M. and A. Juels (2010): On the Impossibility of Cryptography Alone for Privacy-Preserving Cloud Computing. *In Proceedings of the 5th USENIX Workshop on Hot Topics in Security (HotSec)*, 1-8, USENIX Assoc.

Wang, S., D. Agrawal and A. El Abbadi (2012): Is Homomorphic Encryption the Holy Grail for Database Queries on Encrypted Data?, Technical report, Department of Computer Science, UCSB.

Westhoff, D., J. Girao and A. Sarma (2006): Security solutions for wireless sensor networks. *NEC Journal of Advanced Technology* **59**(2): 2-6.

Wu, D. and J. Haven (2012): Using Homomorphic Encryption for Large Scale Statistical Analysis. C. p. Stanford University. Stanford University, CURIS program.

Zhou, H. and G. Wornell (2014): Efficient homomorphic encryption on integer vectors and its applications. *Information Theory and Applications Workshop (ITA)*, 1-9, IEEE.

Zissis, D. and D. Lekkas (2012): Addressing cloud computing security issues. *Future Generation Computer Systems* **28**(3): 583-592.