Security and Privacy Architectures for Biomedical Cloud Computing

Vasiliki Danilatou and Sotiris Ioannidis

***Abstract*— Biomedical research often relies on having access to vast amounts of sensitive information. Patient data in electronic form are held in medical databases and bio-repositories and have to be queried, data mined and operated on by doctors and researchers. Lately, all this information has been migrating to the cloud making access easier for all interested parties. While this helps with dissemination and access, it may have unintended consequences in terms of security and privacy. In this work we propose an architecture that combines distributed access control mechanisms with privacy preserving cryptographic protocols to enable secure sharing and computations on clouds holding sensitive biomedical data. The data shared are tagged with security policies that define who has access to it and how they should be used. Access rights may be delegated to other parties making collaborations easier. Finally, data can be operated on cryptographically to extract specific information without compromising the entire data set.**

I. INTRODUCTION

Patient records, and biomedical data in general, are steadily converted into digital form, and placed in databases and other repositories for ease of access by the appropriate parties. Many countries, healthcare providers, medical practitioners etc. are adopting Electronic Medical Record (EMR) systems, and large companies, such as Google and Microsoft are building medical record clouds (e.g. Google Health and Microsoft HealthVault). A wide range of data relating to a person may appear is such repositories. For example, conditions, medications, allergies, immunizations, procedures, as well as any digital images and files relating to that person. In many cases even DNA sequences are stored on file.

This wealth of information coupled with the open paradigm of online databases and cloud computing, offers tremendous potential for biomedical research, as data can now be easily aggregated and shared. This model however is not without problems primarily relating to the security and privacy of the information. For this reason there are strong regulatory requirements concerning the storage, access and use to such data. For example the Health Insurance Portability and Accountability Act (HIPAA) [1] in the United States places very stringent requirements with respect to privacy in patient records. Also the ISO/TS 18308 standard gives definitions of security and privacy issues for the Electronic Health Record (EHR) [2]

From a practical perspective however we need to come up with architectures that can enable us to work with such sensitive data. Specifically we have the following requirements:

1) Each data owner only controls access to their own data. The data owner may be an individual or an organization.
2) A data owner should be able to grant access to their data to others.
3) Access rights should be delegatable.
4) Any architecture designed for data sharing should scale with the number of data owners.
5) If so desired, computation on data should be done in such a way as to not reveal anything other than the result of the computation.

In this work we present an architecture that addresses the above requirements. Specifically, our architecture accomplishes two broad goals: (i) distributed and scalable access control to private biomedical datasets that can be shared at least partially, and (ii) privacy-preserving computation that only reveals the result of the computation for datasets that cannot be shared.

II. RELATED WORK

The topic of security, access, privacy and anonymity with respect to biomedical data, but also data in general, is a very active research area. We will focus here to works that closely relate to ours. In [11] Zhang and Liu discuss important concepts related to EHR sharing and integration in healthcare clouds, and analyze security and privacy issues in terms of access and management of EHRs. Fung *et al.* in [12] present a survey of of recent advances in privacy-preserving data publishing. In this work we use different building blocks, namely credentials and homomorphic encryption functions, to design a scalable biomedical cloud architecture.

Freedman *et al.* [3] consider the problem of computing the intersection of private datasets of two parties and analyze their protocol under a number of threat models. Their solution is based on representing sets as roots of polynomials. Kissner and Song [10] extend the results of [3] to utilize properties of polynomials beyond evaluation at given points. In this work we utilize these low-level tools in our architecture to support privacy-preserving access to sensitive data in the case of unstrusting parties.

Automated Trust Negotiation was first proposed by Winsborough *et al.* [20]. The purpose is to build trust between participants by having the two parties exchange digitally signed credentials that contain attribute information in order

This work was support in part by the Marie Curie Actions – Reintegration Grants project PASS and by "Hercules" project, no. EL0073, granted under the EEA Financial Mechanism and the Cohesion Fund Managements & Monitoring Special Service (CFM&MSS) Ministry of Economy and Finance Greece.

Vasiliki Danilatou is with the Institute of Molecular Biology and Biotechnology Foundation for Research and Technology - Hellas vdanil@imbb.forth.gr

Sotiris Ioannidis is with the Institute of Computer Science, Foundation for Research and Technology - Hellas sotiris@ics.forth.gr

to establish trust and make access control decisions. In environments like the Internet where there may be few or no pre-existing relationships, parties that seek to form a trust relationship might be unwilling to release sensitive credentials [13], [17]. To address this difficulty, a number of schemes have been proposed recently that use cryptography or multiple rounds of negotiations to protect credentials and attributes [14], [15], [16], [18], [19]. Using an automated trust negotiation scheme, participants specify access control policies for the disclosure of credentials. They then enter a negotiation phase which consists of a sequence of exchanges that are controlled by the access control policies defined for the credentials. At each round, parties gain higher levels of mutual trust, permitting access control policies for more sensitive credentials to be satisfied, which in turn enable these credentials to be exchanged. In our work we assume the date owner as well as the data user have a well known trust relationship, and credentials are issued that implement these policies. In the case of lack of trust, we use cryptographic privacy-preserving protocols.

The STRONGMAN system described in [22] demonstrates three new approaches to providing efficient local policy enforcement complying with global security policies. First is the use of a compliance checker to provide great local autonomy within the constraints of a global security policy. Second is a mechanism to compose policy rules into a coherent enforceable set, *e.g.,* at the boundaries of two locally autonomous application domains. Third is the "lazy instantiation" of policies to reduce the amount of state enforcement points need to maintain. In this work we utilize the KeyNote system which was developed as part of STRONGMAN to build strong and scalable access control is biomedical data clouds.

III. Architecture

A. Preliminaries

We define the participants in our architecture to be (i) the biomedical data owners that hold their data in network accessible repositories or clouds, and (ii) the data consumers that seek access to the data. A data owner can also be a data consumer and vice versa. Sharing of sensitive information between participants is a very challenging task. Each participant may have their own data representation and their own security policies in place. For the purpose of this work we assume that anyone that wants to participate in data sharing must use the same data format and agree on the same data operations. Different data items are represented as *unique* natural numbers. The same number cannot be used to represent two different data items. For example if the number 12345 is used to represent medication XYZ, it cannot be used to also represent disease ZYX. Our architecture can be agnostic to the type of data, which may range from medication XYZ, it cannot be used to also represent disease ZYX. Our architecture can be agnostic to the type of data, which may range from simply patient names to DNA sequences, as anything can be represented by a number or bitstring.

```
Local-Constants:
  BIOMEDICAL_DATAOWNER_KEY="rsa-base64:M..."
  PHYSICIAN_KEY KEY = "rsa-base64:MIGJAo..."
Authorizer: BIOMEDICAL_DATAOWNER
Licensees: PHYSICIAN
Conditions:
  ((app_domain == "CLOUD_COMPUTING") &&
   (records == "ALL_PATIENTS") &&
   (permissions == "READ_ACCESS") &&
   (valid <= "20101231")) -> "permit";
Signature: "sig-rsa-sha1-base64:QU6..."
```

Fig. 1. Sample credential for allowing a physician to read patient records off a biomedical data cloud.

```
Local-Constants:
  PHYSICIAN_KEY="rsa-base64:McgFJX..."
  RESEARCHER_KEY = "rsa-base64:MCgQGB..."
Authorizer: PHYSICIAN
Licensees: RESEARCHER
Conditions:
  ((app_domain == "CLOUD_COMPUTING") &&
   (records == "CANCER_PATIENTS") &&
   (record_state == "ANONYMIZED") &&
   (permissions == "READ_ACCESS") &&
   (valid <= "20101130") -> "permit";
Signature: "sig-rsa-sha1-base64:Qpf..."
```

Fig. 2. Sample credential for delegating some access rights from the physician to a collaborating researcher.

We also assume that each participant will use a semantically secure public-key cryptosystem and generate the corresponding public and private keys.

B. Distributed Access Control

The main architectural principle behind our system is the ability of data owners to issue access *credentials* to data users. Data users in turn may further issue access credentials to other data users. A credential is a statement that specifies what access rights it's holder has with respect to very specific data. The credential is cryptographically signed by it's issuer. The holder of the credential may present it to the issuer to gain access to the data.

Let us examine how the process works with an example. Figure 1 shows a simple credential. For the purpose of this work we are using the KeyNote Trust Management system [5], [6] which provides us with the necessary credential functionality. The credential in our example is issued by a biomedical data owner to a physician, granting read access to all of the data owner's patient records. The credential contains the public keys of the two parties along with the cryptographic signature (computed by the authorizer) verifying the validity of the credential. The specific credential also has an expiration date, invalidating it past that date. Finally the credential has an extra field specifying the type of application the credential is supposed to be used for, in this case cloud computing.

The physician may now present this credential to the data owner every time they want to read a patient record. The physician may also issue new credentials for collaborators. In our example in Figure 2, the physician creates a new credential and grants read access to only the cancer patient records that have been anonymized, and for a shorter period of time. The physician then signs the credential and gives it to the researcher along with the original credential that the data owner issued to the physician. Essentially this creates a chain of credentials, all cryptographically signed by their issuer.

The researcher can now go directly to the data owner, present this chain of credentials, which can in turn be cryptographically verified by the data owner by checking the correctness of the signatures. If all the conditions are met, and the signature are valid, then access is granted.

C. Scalability

Using credentials to hand out access rights is ideal for distributed environments as they remove the bottleneck of managing access rights centrally and the cumbersome use of logins and passwords [7], [9], [8]. Credentials wrap the authentication and authorization procedure in on cryptographically signed token.The credential itself is sufficient to prove the validity of the access. Additionally, the ability to delegate access rights further helps in offloading the management burden. Someone that has valid rights to access certain data, may issue credentials to their collaborators without involving the data owner. Delegation is always hierarchical, and no one can escalate their access rights that can lead to a security and privacy breach. That is, the permissions one can grant are always a subset of the permissions they hold.

D. Privacy Discussion

So far we have primarily covered the case of security and scalability, and to a lesser degree privacy. While sharing is controlled in terms of who has a certain type of access to a certain type of data, once this access is given, the data can at the very least be looked at. What happens in cases where we don't want to reveal any data prior to the execution of the computation. For example, assume that two participants want to discover whether they have any common patients with a specific disease. How can they do this without revealing their entire patient list along with their corresponding diseases.

E. Cryptographic Tools

The basic tool we use for computation in the case of mistrust between parties is a privacy-preserving set intersection protocol. Specifically we use work done in [3], by Freedman *et al.* Their private matching scheme is a two-party protocol between a client C and a server S. When the protocol starts, both parties have private data sets (Z_C and Z_S) drawn from some common domain. At the conclusion of the protocol, the chooser learns the intersection $Z_C \cap Z_S$, but nothing about any other data in Z_S. That is, Freedman *et al.* prove that their protocol is privacy-preserving in the semi-honest model. For data sets of size $O(k)$, the protocols results in a communication overhead of $O(k)$ and computational overhead of $O(k \ln \ln k)$.

The Freedman protocol is based on a semantically secure homomorphic encryption scheme: If E is a public encryption function of a homomorphic encryption scheme, then, given the ciphertexts $c_1 = E(m_1)$ and $c_2 = E(m_2)$, the ciphertext $c^* = E(m_1 + m_2)$ can be computed efficiently without knowledge of the private key. Similarly, given $c = E(m)$ and some r from the group of plaintexts, then $c^* = E(rm)$ can be computed efficiently without knowledge of the private key. A public encryption function E is semantically secure if it is computationally infeasible for an attacker to derive significant information about a plaintext given only its ciphertext and the public encryption key. An example of a semantically secure homomorphic encryption scheme is Paillier's cryptosystem [4]. The homomorphic property of an encryption function E implies that anyone in possession of the encrypted coefficients of a polynomial $f(x)$ can compute a valid encryption of $f(y)$ for any y from the group of plaintexts without the knowledge of the private key or the coefficients. In particular, for any known plaintexts y_1, y_2 and any known constant r, a valid encryption $E(rf(y_1)+y_2)$ can be computed without the knowledge of the private key or the coefficients of $f(x)$.

F. Example Private Computation between Mistrusting Parties

Continuing the discussion from Section III-D and given the tools from Section III-E we can see that biomedical data owners can share their data under certain conditions, even when they do not entirely trust the requester. Specifically, if two parties are willing to at least reveal the data they have in common, it is possible to do it, and guarantee that no other data is revealed. This is very often the case in medical research, where both parties can benefit from exploring each others dataset. After all, if the same data exist in both datasets, is was known to both parties to begin with, and no privacy constraints were violated.

To see how this works in practice, consider two biomedical data owners (let us call them Alice and Bob) that want to conduct some collaborative research, but no one of the two wants to give (even anonymized) access to their dataset to the other. The use of credentials as we described in the first part of our architecture is therefore insufficient. Assume that Alice's dataset is $Z_A = a_1, ..., a_n$ and Bob's dataset is $Z_A = b_1, ..., b_n$. Remember that a_i and b_i need to be drawn from the same domain (see Sections III-A and III-E). Also, let us define a semantically secure public-key homomorphic encryption scheme $S = (E_{pk}, D_{sk})$ e.g. Paillier's cryptosystem [4]. That is was have generated a public key which we use for encryption and a private key which we use for decryption.

Alice starts by creating a polynomial:

$$f(x) = (x - a_1)(x - a_2) \cdots (x - a_n) = \sum_{i=0}^{n} \alpha_i x^i$$

She then encrypts each coefficient α_i $(i = 1, \ldots, n)$ under E_{pk} getting $E_{pk}(\alpha_i)$ and proceeds to send those encrypted coefficients to Bob. Bob then proceeds to compute $E_{pk}(f(b_i))$ for every $b_i \in Z_B$. Note, that Bob, cannot simply use the public key to compute the these values, because even though the public key is well know, the polynomial is secret. Bob, can however compute the encrypted polynomial using the properties of the homomorphic encryption scheme (see Section III-E). For every $b_i \in Z_A$ then $f(b_i) = 0$, that is data of Bob that also exist in Alice's dataset are roots of the polynomial Alice computed. As Bob does not want to reveal any additional information other that what is in the common dataset, he randomizes all his encryptions by a random, non-zero, value r. This is done by using the properties presented in Section III-E, $E_{pk}(f(b_i))^r = E_{pk}(r\dot{f}(b_i))$. If $f(b_i) = 0$ then obviously the encryption of $E_{pk}(r\dot{f}(b_i)) = E_{pk}(0)$, otherwise it it some random value. Bob must however provide some information to Alice to check whether some of Bob's data exist in her dataset. To do this Bob computes the following: $E_{pk}(r\dot{f}(b_i) + b_i)$ and sends it to Alice. Alice decrypts it using her secret key: $D_{sk}(E_{pk}(r\dot{f}(b_i) + b_i))$. The resulting plaintext data will be b_i if and only if $b_i \in A$.

1) Security Analysis: Obviously, one of the two parties may lie in the interaction by supplying false data. For example, one can enumerate all possible numbers to try to extract the others dataset. This is the age old problem of dataset extraction which may be addressed by out-of-band methods. For example limiting the data that can be queried at every interaction, or denying to further interact with parties that have been found to lie. One may actually go a step further and add a third component to the so far discussed architecture. That is extend it with a reputation system, that measures the trustworthiness of participants [21].

IV. Conclusions

In this work we presented a two-tier architecture for security and privacy in biomedical clouds. We combined the power of decentralized management and access control, provided by cryptographic credentials, with the ability to perform privacy-preserving set operations on data.

The first part of our architecture enables biomedical data owners to easily hand out access to physicians, researchers, etc. They in turn, may delegate further access to their collaborators. Of course, even though such an approach provided great flexibility in terms of sharing information, it is insufficient on its own when we would like to avoid revealing information unnecessarily. For this reason we combine cryptographic credentials with privacy-preserving protocols. Privacy-preserving protocols permit untrusting parties to perform specific operations without revealing the entire dataset but only the result of the operation. This is particularly important in biomedical research, and biomedicine in general. Now organizations may datamine each other datasets for common patterns when for example they perform research on diseases or experimental drugs. They can do this without revealing extraneous information.

References

[1] Centers for Medicare and Medicaid Services. The Health Insurance Protability and Accountability Act of 1996 (HIPPA) http://www.cms.gov/HIPAAGenInfo/

[2] ANSI. ISO/TS 18308 Health Informatics Requirments for an Electronic Health Record Architecture, ISO 2004.

[3] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In *EUROCRYPT*, volume 3027 of *LNCS*, pages 1–19, 2004.

[4] P. Paillier. Public-key Cryptosystems Based on Composite Degree Residuosity Classes. In *Proceedings of EUROCRYPT'99*, 1999.

[5] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. The role of trust management in distributed systems security. In *Secure Internet Programming*, volume 1603 of *Lecture Notes in Computer Science*, pages 185–210. Springer-Verlag Inc., New York, NY, USA, 1999.

[6] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis. The KeyNote Trust Management System Version 2. RFC 2704, September 1999.

[7] Sotiris Ioannidis, Steven M. Bellovin, John Ioannidis, Angelos D. Keromytis, and J.M. Smith. Design and implementation of virtual private services. In *Proceedings of the IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), Workshop on Enterprise Security, Special Session on Trust Management in Collaborative Global Computing*, June 2003.

[8] Sotiris Ioannidis, Steven M. Bellovin, John Ioannidis, Angelos D. Keromytis, Kostas Anagnostakis and J.M. Smith. Virtual Private Services: Coordinated Policy Enforcement for Distributed Applications In *International Journal of Network Security (IJNS) 4(3):69–80*, January 2007.

[9] S. Ioannidis, A.D. Keromytis, S.M. Bellovin, and J.M. Smith. Implementing a Distributed Firewall. In *Proceedings of Computer and Communications Security (CCS) 2000*, pages 190–199, November 2000.

[10] L. Kissner and D. Song. Private and Threshold Set-Intersection. In *Proceedings of CRYPTO'05*, 2005.

[11] Rui Zhang and Ling Liu. Security Models and Requirements for Healthcare Application Clouds. In *Proceedings of the 3rd IEEE International Conference on Cloud Computing (Cloud2010)*. July 2010.

[12] B.C.M. Fung, K. Wang, R.Chen and P.S. Yu. Privacy-preserving data publishing: A survey of recent developments. In *ACM Computing Surveys (CSUR) 42(4)*, December 2010.

[13] W. Winsborough and N. Li. Towards Practical Automated Trust Negotiation. In *Proceedings of IEEE 3rd International Workshop on Policies for Distributed Systems and Networks (Policy'02)*, 2002.

[14] W. H. Winsborough and N. Li. Protecting Sensitive Attributes in Automated Trust Negotiation. In *Proceedings of the 2002 ACM Workshop on Privacy in the Electronic Society (WPES'02)*, 2002.

[15] W. H. Winsborough and N. Li. Safety in Automated Trust Negotiation. In *IEEE Symposium on Security and Privacy (S&P'04)*, 2004.

[16] J. E. Holt, R. W. Bradshaw, K. E. Seamons, and H. Orman. Hidden Credentials. In *WPES'03: Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society*, 2003.

[17] K. Irwin and T. Yu. Preventing Attribute Information Leakage in Automated Trust Negotiation. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS'05)*, 2005.

[18] K. Seamons, M. Winslett, and T. Yu. Limiting the Disclosure of Access Control Policies during Automated Trust Negotiation. In *Proceedings of Network and Distributed System Security Symposium*, 2001.

[19] K. E. Seamons, M. Winslett, T. Yu, L. Yu, and R. Jarvis. Protecting Privacy During On-line Trust Negotiation. In *Proceedings of the 2nd Workshop on Privacy Enhancing Technologies (PET'02)*, 2002.

[20] W. H. Winsborough, K. E. Seamons, and V. E. Jones. Automated Trust Negotiation. In *Proceedings of DARPA Information Survivability Conference and Exposition*, 2000.

[21] Andrew G. West, Adam J. Aviv, Jian Chang, Vinayak S. Prabhu, Matt Blaze, Sampath Kannan, Insup Lee, Jonathan M. Smith and Oleg Sokolsky. QuanTM: A Quantitative Trust Management System. In *Proceedings of the European Workshop on System Security (EUROSEC)*, 2009.

[22] Angelos D. Keromytis, Sotiris Ioannidis, Michael B. Greenwald, and Jonathan M. Smith. The STRONGMAN Architecture. In *DARPA Information Survivability Conference and Exposition (DISCEX III)*, pages 178–188. IEEE Computer Society Press, April 2003.