HealthShare: Using Attribute-Based Encryption for Secure Data Sharing Between Multiple Clouds

Michalas, A. and Weingarten, N.

https://dx.doi.org/10.1109/CBMS.017.30

HealthShare: Using Attribute-Based Encryption for Secure Data Sharing Between Multiple Clouds

Antonis Michalas
Cyber Security Group,
Department of Computer Science
University of Westminster, UK
a.michalas@westminster.ac.uk

Noam Weingarten
Cyber Security Group,
Department of Computer Science
University of Westminster, UK
N.Weingarten01@westminster.ac.uk

***Abstract*—In this invited paper, we propose HealthShare – a forward-looking approach for secure ehealth data sharing between multiple organizations that are hosting patients' data in different clouds. The proposed protocol is based on a Revocable Key-Policy Attribute-Based Encryption scheme and allows users to share encrypted health records based on a policy that has been defined by the data owner (i.e. patient, a member of the hospital, etc). Furthermore, access to a malicious or compromised user/organization can be easily revoked without the need to generate fresh encryption keys.**

***Keywords*—eHealth; Security; Cloud Computing; EHR Protection; Access Control; Policies; Attribute-Based Encryption;**

I. Introduction

Not many years ago, eHealth was seen as an expenditure rather than an investment. During the last decade this has changed significantly, to the extent that eHealth has moved to the top of the development agenda not only for private organizations but also for public administration bodies that have spurred the development of eHealth. To this end, we have seen a steady increase in research focus and funding aiming to modernize existing healthcare systems and to provide reliable and cost effective eHealth services [1]. As a result, nowadays we are faced with a major technological upturn of an industry that for many years has relied on handwritten records and now is expanding at a phenomenal rate.

As adoption of ehealth solutions advances, new computing paradigms - such as cloud computing - bring the potential to improve efficiency in managing medical health records, help reduce costs [2] and support the collaboration between different organizations. However, placing patients' data in remote locations raises many concerns regarding the privacy of users' data.

Lately we have seen some significant developments in cloud computing [3]. Researchers have been focusing not only on creating efficient and flexible cloud-based services but a lot of attention is being invested in designing systems that are secure against various malicious behaviours and attacks. As a result, many companies and individuals have been starting using cloud-based services with main aim to improve their productivity as well as the collaboration between the users.

When it comes to the health industry, many companies and organizations have already migrated their services to the cloud. In addition to that, many of the existing cloud-based services are enhanced with security-related mechanisms that provide the necessary guarantees for the protection of patients' data. However, there is a clear set of mechanisms that will allow different organizations (e.g. hospitals) to host patients' data in different clouds in order to securely share health records. Having such mechanisms can increase the productivity of health practitioners since a patients medical records can be easily transferred to a different hospital. In addition to that, such a sharing functionality has the potential to better support research and enhance the collaboration between specialists and scientists that are geographically apart. This can be exhibited by doctors who have certain specialities can be granted rights to be able to examine records of patients that exhibit certain symptoms. The ease of transferring patients' medical records between hospitals will enable the transferring of patients' care from one hospital to another will alleviate the requirements to repeat investigations.

A. Our Contribution

In this invited paper, we propose HealthShare – a forward-looking approach for secure ehealth data sharing between multiple organizations that are hosting patients' data in different clouds. The proposed protocol is based on a Revocable Key- Policy Attribute-Based Encryption scheme and allows users to share encrypted health records based on a policy that has been defined by the data owner (i.e. patient, a member of the hospital, etc). Furthermore, access to a malicious or compromised user/organization can be easily revoked without the need to generate fresh encryption keys. We hope that this work will give valuable insights to the designers of cloud-based eHealth services in order to help design and develop mechanisms that will support data sharing in a multi-cloud environment.

B. Organization

The remainder of this paper is organized as follows: In Section II, we present important works that focus on secure cloud storage for ehealth services and privacy-preserving data sharing. In Section III, we present the main entities that participate in our model and we proceed by defining the problem statement while in Section IV, we describe the cryptographic primitives that are needed for a proper run of the protocol.

In Section V, we introduce the threat model that we will consider and in Section VI we describe our protocol. Finally, in Section VII we conclude the paper.

II. Related Work

In this section we present the related works that mainly focus on the problem of secure data sharing in a cloud environment.

In [4] authors presented DBSP – a framework for data and operation security in Infrastructure-as-a-Service (IaaS) clouds, consisting of protocols for a trusted launch of virtual machines and domain-based storage protection. Its security guarantees are supported by an extensive theoretical analysis with proofs about protocol resistance against attacks in the defined threat model. The protocols allow trust to be established by remotely attesting host platform configuration prior to launching guest virtual machines and ensure confidentiality of data in remote storage, with encryption keys maintained outside of the IaaS domain. In addition to that, authors provide functionality for sharing data between different domains. To this end, they present an XML-based language framework which enables clients of IaaS clouds to securely share data and clearly define access rights granted to peers. The paper also presents experimental results that demonstrate the validity and efficiency of the proposed protocols. The experimental results are based on an implementation of DBSP as an extension of OpenStack, a popular open-source cloud-computing platform. Even though the sharing functionality proposed in DBSP is based on standard cryptographic primitives, which makes it rather efficient, it is also considered as basic. In addition to that, the main drawback is the fact that DBSP is using a symmetric key to encrypt an entire disk. As a result, to give access to a user, data owner must reveal the secret key.

Authors in [5] proposed a protocol that allows practitioners to identify duplicate data in privacy-preserving way. More precisely, the protocol is efficient and scalable for practical uses and allows health practitioners to remove duplicate records for a patient without learning anything about the content of the records. Moreover, authors conducted extensive experimental results by using real health records. While the protocol is focusing on a different problem, there are some similarities with ours since in both cases health data are somehow shared in a privacy-preserving way.

In [6] authors presented a forward-looking design for secure storage and file sharing in cloud environments. The scheme was based on a Symmetric Searchable Encryption (SSE) scheme [7], [8], [9] that allows patients of an electronic healthcare system to securely store encrypted versions of their medical data and search directly on them without having to decrypt them first. Even though the scheme offers some kind of secure sharing it is not that flexible and efficient since it does not rely on policies. Furthermore, even though authors provide a discussion regarding access revocation they do not provide a concrete and efficient solution. Hence, the protocol is considered as inefficient for sharing large amount of data between multiple users.

In [10], author showed how to construct a framework for secure file sharing by using the benefits of Revocable Attribute-Based Encryption. More precisely, the protocol is using a Key-Policy Attribute-Based technique through which access revocation is optimized. Moreover, author showed how to securely and efficiently remove access to a file, for a certain user that is misbehaving or is no longer part of a user group, without having to decrypt and re-encrypt the original data with a new key or a new policy.

III. System Model and Problem Statement

In this section, we introduce the system model that we consider by explicitly describing the main entities that participate in our protocol. The system model of our work is built on top of the model presented in [10]. Furthermore, we strictly define the problem statement.

Cloud Service Provider (CSP): One of the common models of a cloud computing platform is Infrastructure-as-a-Service (IaaS). In its simplest form, such a platform consists of cloud hosts which operate virtual machine guests and communicate through a network. Often a cloud middleware manages the cloud hosts, virtual machine guests, network communication, storage resources, a public key infrastructure and other resources. Cloud middleware creates the *cloud infrastructure* abstraction by weaving the available resources into a single platform. In our system model we consider a cloud computing environment based on a trusted IaaS provider like the one described in [11]. The IaaS platform consists of cloud hosts which operate virtual machine guests and communicate through a network. In addition to that, we assume a Platform-as-a-Service (PaaS) provider, like the one described in [8], [12], that is built on top of the IaaS platform and can host multiple outsourced databases. Furthermore, the cloud service provider is responsible for storing the data of users and also providing data access. Furthermore, for the needs of this work we assume that different organizations using different CSP's. To this end, we denote $\mathcal{CS} = \{cs_1, \ldots, cs_m\}$ be the set of all available cloud service providers that are hosting an eHealth service.

Registration Authority (RA): RA is responsible for the registration of users. Additionally, RA has a public/private key pair denoted as $\mathsf{pk}_{\mathsf{RA}}/\mathsf{sk}_{\mathsf{RA}}$. Apart from that, RA is responsible for generating parameters that will be used for the proper function of the application (e.g. reveal the identity of a misbehaving user). RA can run as a separate third party but can be also implemented as part of the cloud platform we described earlier.

a) Master Authority (MA): The master authority has a master secret key MSK and a public key pk. The master key is kept private while the public key is known to everyone. Additionally, MA uses MSK to generate private keys for new users.

User (u): In our scenario a user interacts with the CSP in order to mange certain files that has access to. The operations that a user can perform are the following: *a)* register to the service, *b)* generate encryption keys to safely protect her data, *c)* store data in the cloud, *d)* share data with other users by creating certain policies using a Key Policy Attribute-Based Encryption scheme. Furthermore, each user has a unique identifier u_i. A user u_i might be also referred as data owner when she is the one who generated a certain file. Each user u_i has a private/public key pair ($\mathsf{pk_i}/\mathsf{sk_i}$). The private key is kept secret, while the public key is shared with the rest of the community. These keys will be used to secure message exchanges in the community, hence the communication lines between parties are assumed to be secure. It is also assumed that users knows the public keys of RA, MA and the hosts operated by the CSP.

Problem Statement: *Let $\mathcal{CS} = \{cs_1, \ldots, cs_m\}$ be the set of all available cloud service providers that are hosting an eHealth service. Furthermore, let $\mathcal{U}_i = \left\{u_1^i, \ldots, u_n^i\right\}$ be the set of all users that are registered through a registration authority (RA) to a health service hs_i and $\mathcal{U}_j = \left\{u_1^j, \ldots, u_n^j\right\}$ be the set of all users that are registered to a health service hs_j. Lets assume that a user u_k^i stores a file m to the local storage of hs_i. The problem here is to find a way to achieve the following:*

1. *Keep the content of each m private against both internal and external attacks;*
2. *User u_k^i should be able to securely share m with another user $u_l^j \in \mathcal{U}_j$ who has access to hs_j based on a certain policy;*
3. *Data owner u_k^i should be able to efficiently revoke access to a user u_l^j for a file that has shared with her. This should not require the data owner to decrypt and re-encrypt the file with a fresh key.*

IV. Cryptographic Primitives

In this section, we introduce the notations that we use throughout the rest of the paper as well as the threat model that we consider.

A. Notation

In order to provide a concrete and reliable solution for the problem described in Section III, we need to build a protocol through which newly encrypted data will not be decryptable by a user if that user's access has been revoked. In addition to that, we want to allow users with certain access rights to be able to search directly over encrypted data. To this end, we will be using a key-policy attribute-based encryption (KP-ABE) scheme such as the one described in [10]. In a KP-ABE scheme every secret key is generated with a policy P and every ciphertext is bound to a set of attributes U. Then, decryption is only possible if $P(U) = \mathsf{True}$. From now on we will refer to the set of all available attributes as $\Omega = \{a_1, \ldots, a_n\}$, while the set of all available policies will be denoted as $\mathcal{P} = \{P_1, \ldots, P_m\}$.

TABLE I
Notation index

Symbol	Description
CSP	Cloud Service Provider
RA	Registration Authority
u_i	A user with unique identifier i
m	An arbitrary message
Enc	Encryption algorithm
Dec	Decryption algorithm
$\mathsf{pk_i}/\mathsf{sk_i}$	Public/private key pair of user i
MSK	A master secret key of MA i
a	Attribute
P	Policy
rl	Revocation List

We now proceed with the definition of a revocable KP-ABE scheme as described in [13].

Definition 1 (Revocable Key-Policy ABE): A revocable KP-ABE scheme is a tuple of the following five algorithms:

1. Setup is a probabilistic algorithm that takes as input a security parameter λ and outputs a public key pk and a master key MSK. We denote this by $(\mathsf{pk}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$.
2. Gen is a probabilistic algorithm that takes as input a master key, a policy $P \in \mathcal{P}$ and the unique identifier of a user and outputs a secret key which is bind both to the corresponding policy and user. We denote this by $(\mathsf{sk_{P,ID}}) \leftarrow \mathsf{Gen}(\mathsf{MSK}, P, ID)$.
3. Enc is a probabilistic algorithm that takes as input a public key, a message m, a set of attributes $\mathcal{S} \in \Omega$ and a timestamp t. After a proper run, the algorithm outputs a ciphertext $c_{\mathcal{S},t}$ which is bind both to the set of attributes and the time. We denote this by $(\mathsf{c_{\mathcal{S},t}}) \leftarrow \mathsf{Enc}(\mathsf{pk}, m, \mathcal{S}, t)$.
4. $\mathsf{KeyUpdate}$ is a probabilistic algorithm that takes as input a master key, a revocation list rl and a timestamp t and outputs a key update information for time t. We denote this by $(\mathsf{K_t}) \leftarrow \mathsf{KeyUpdate}(\mathsf{MSK}, rl, t)$.
5. Dec is a deterministic algorithm that takes as input a secret key, a key update $\mathsf{K_{t'}}$ and a ciphertext and outputs the original message m *iff* the set of attributes $\mathcal{S}$ that are bind to the ciphertext satisfies the policy P, $t' \geq t$ and the ID of the corresponding user was not revoked at time t. We denote this by $\mathsf{Dec}(\mathsf{sk_{P,ID}}, \mathsf{K_{t'}}, c_{\mathcal{S},t}) \rightarrow m$.

A summary of the notation introduced so far is presented in Table I.

V. Threat Model

Our threat model is similar with the one described in [4], which is based on the Dolev-Yao adversarial model [14] and

further assumes that privileged access rights can used by a remote adversary $\mathcal{ADV}$ to leak confidential information. $\mathcal{ADV}$, e.g. a corrupted system administrator, can obtain remote access to any host maintained by the IaaS provider, but cannot access the volatile memory of guest VMs residing on the compute hosts of the IaaS provider.

Hardware Integrity: We assume that the cloud provider has taken all the necessary technical and non-technical measures in order to protect the underling hardware from tampering.

Physical Security: We assume physical security of the data centres where the IaaS is deployed. This assumption holds both when the IaaS provider owns and manages the data center (as in the case of Amazon Web Services, Google Compute Engine, Microsoft Azure, etc.) and when the provider utilizes third party capacity, since physical security can be observed, enforced and verified through known best practices by audit organizations. This assumption is important to build higher-level hardware and software security guarantees for the components of the IaaS. We assume the record is kept on protected storage with read-only access and the adversary cannot tamper with it.

Network Infrastructure: The IaaS provider has physical and administrative control of the network. $\mathcal{ADV}$ is in full control of the network configuration, can overhear, create, replay and destroy all the exchanged messages between the CSP and their resources (virtual machines, database components etc) as well as with other entities that participate in our system model (i.e. the registration authority).

Cryptographic Security: We assume encryption schemes are semantically secure and the $\mathcal{ADV}$ cannot obtain the plain text of encrypted messages. In addition to that, we explicitly assume that the $\mathcal{ADV}$ cannot forge the revocation list and cannot decrypt a ciphertext without knowing the corresponding secret key. Furthermore, we assume that the probability of $\mathcal{ADV}$ guessing a generated random number is negligible. Finally, we explicitly exclude denial-of-service attacks [15], [16], [17], [18] from our adversarial model and we focus on $\mathcal{ADV}$ that aims to compromise the confidentiality of data by forging existing access policies generated by the corresponding data owners.

VI. Protcol Description (HealthShare)

In this section, we present HealthShare that constitutes the core of this paper's contribution. Since this is a position paper, we will not present a formal construction of the protocol. We will provide provide though a detailed high level description (also see figure 1) that gives the reader a good overview of the functionality that is offered as well as a typical use-case scenario.

A user $u_k^i \in \mathcal{U}_i$ registers to an electronic healthcare service hs_i through the registration authority. After the successful registration registration, u_k^i will receive a credential that can use in order to store and retrieve data to the CSP cs_i that hs_i is using.

For simplicity, from now on when we refer to a user we will be assuming that user is a patient. However, in our protocol the group of users is comprised by both patients and healthcare practitioners or any entity that has the right to store data in the CSP. Now that u_k^i has registered, she can start uploading files to the CSP. Hence, we assume that u_k^i can store the results m of an exam on the remote storage offered by cs_i. To do that in a secure and privacy-preserving way the files needs to be transmitted and stored in an encrypted form in order to avoid both internal and external attacks. To this end, u_k^i contacts MA who generates an ABE key based on a certain policy. Now that u_k^i received her ABE secret key she can start encrypting files. Hence, u_k^i encrypts m by using a set of attributes defined by her. The attributes can be seen as access rights since only users with keys that their policy satisfies these attributes will be able to decrypt the data. The generated ciphertext is sent to the CSP who cannot decrypt it since it does not have knowledge of a valid private key – hence the content of the file remains private even if the CSP acts maliciously.

Now that u_k^i has stored a file in the cs_i, she wishes to share it with a user $u_l^j \in \mathcal{U}_|$ who is registered at sh_j and has access to cs_j. To do this, u_l^j generates an ABE key by contacting MA. This key is bound to a certain policy like before. However, it worth mentioning that the time this key is generated, the generation process is irrelevant to the file that u_l^j wishes to access. Now that u_l^j has generated her ABE key she can ask to access the encrypted version of m (c_m) that stored earlier by u_k^i. this requires that cs_i and cs_j are sharing a search function that allows users to look for stored files in both locations. As a next step, u_l^j will receive c_m from cs_i and will try to decrypt it. However, the decryption will only work if the attributed that are bound to c_m are satisfying the policy that is attached on u_l^j's key. In any other case the decryption will fail and contents of m will remain private.

Apart from successfully sharing files, one of our goals is to efficiently revoke access for a user. Having this in mind, we assume that u_k^i wishes to revoke access for the users u_l^j. To do so, u_k^i will only have to run an algorithm that will actually revoke access for the unique key that was generated for user u_l^j. Apart from that, u_k^i will not have to decrypt and then re-encrypt the file with a fresh key since the key that is hold by u_l^j will not be a valid decryption key.

VII. Conclusion

In this invited paper, we proposed a protocol for secure and efficient sharing of ehealth data in a multi-cloud environment. The proposed protocol was based on a Revocable Key-Policy Attribute-Encryption and allowed users to control access rights directly on encrypted data. Moreover, the proposed protocol allows secure and efficient revocation of access to data, for a certain user that is compromised, misbehaving or is no longer part of a user group, without having to decrypt and re-encrypt the original data.

As future steps, we plan to implement our protocol in order to measure its performance and prove its effectiveness in a real

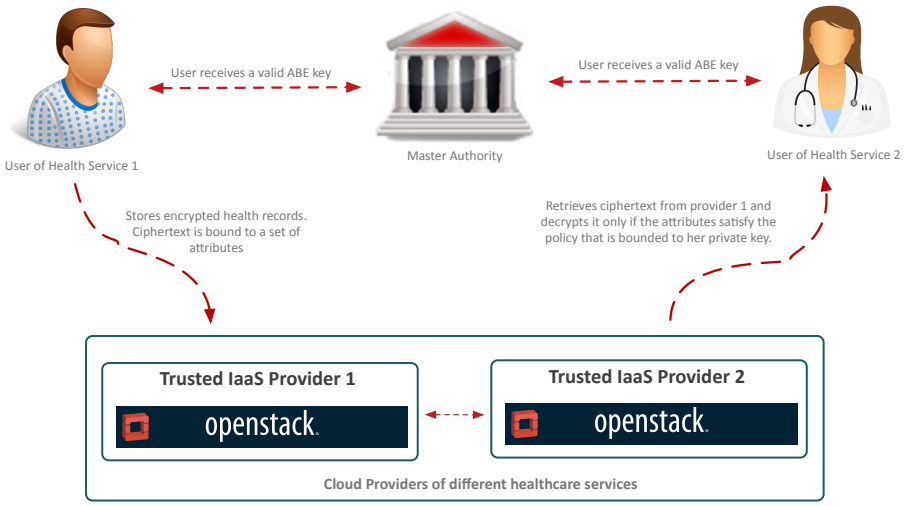

Fig. 1. HealthShare overview

cloud environment. Furthermore, we plan to explore the incorporation of our protocol with mobile sensing applications and with privacy-preserving reputation systems for cloud-based participatory sensing applications. The envisioned system will be based on [19], [20], [21] and will effectively maintain the privacy and anonymity of users [22], [23].

REFERENCES

[1] K. Yigzaw, A. Michalas, and J. Bellika, "Secure and scalable statistical computation of questionnaire data in r," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2016.

[2] A. Michalas, N. Paladi, and C. Gehrmann, "Security aspects of e-health systems migration to the cloud," in *e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on*, pp. 212–218, IEEE, 2014.

[3] A. Michalas and M. Bakopoulos, "Secgod google docs: Now i feel safer!," in *2012 International Conference for Internet Technology And Secured Transactions*, pp. 589–595, Dec 2012.

[4] N. Paladi, C. Gehrmann, and A. Michalas, "Providing user security guarantees in public infrastructure clouds," *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1–1, 2016.

[5] K. Y. Yigzaw, A. Michalas, and J. G. Bellika, "Secure and scalable deduplication of horizontally partitioned health data for privacy-preserving distributed statistical computation," *BMC Medical Informatics and Decision Making*, vol. 17, no. 1, p. 1, 2017.

[6] A. Michalas and R. Dowsley, "Towards trusted ehealth services in the cloud," in *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*, pp. 618–623, Dec 2015.

[7] R. Dowsley, A. Michalas, and M. Nagel, "A report on design and implementation of protected searchable data in iaas," tech. rep., Swedish Institute of Computer Science (SICS), 2016.

[8] Y. Verginadis, A. Michalas, P. Gouvas, G. Schiefer, G. Hbsch, and I. Paraskakis, "Paasword: A holistic data privacy and security by design framework for cloud services," in *Proceedings of the 5th International Conference on Cloud Computing and Services Science*, pp. 206–213, 2015.

[9] A. Michalas and K. Y. Yigzaw, "Locless: Do you really care your cloud files are?," in *2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC)*, pp. 618–623, Dec 2015.

[10] A. Michalas, "Sharing in the rain: Secure and efficient data sharing for the cloud," in *2016 International Conference for Internet Technology And Secured Transactions*, pp. 589–595, Dec 2016.

[11] N. Paladi, A. Michalas, and C. Gehrmann, "Domain based storage protection with secure access control for the cloud," in *Proceedings of the 2014 International Workshop on Security in Cloud Computing*, ASIACCS '14, (New York, NY, USA), ACM, 2014.

[12] Y. Verginadis, A. Michalas, P. Gouvas, G. Schiefer, G. Hübsch, and I. Paraskakis, "Paasword: A holistic data privacy and security by design framework for cloud services," pp. 1–16, 2017.

[13] A. Sahai and H. Seyalioglu, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *in Proceedings of the 32nd Annual International Cryptology Conference: Advances in Cryptology - CRYPTO2012*, pp. 199–217, Springer, 2012.

[14] D. Dolev and A. C. Yao, "On the security of public key protocols," *Information Theory, IEEE Transactions on*, vol. 29, no. 2, 1983.

[15] A. Michalas, N. Komninos, N. R. Prasad, and V. A. Oleshchuk, "New client puzzle approach for dos resistance in ad hoc networks," in *Information Theory and Information Security (ICITIS), 2010 IEEE International Conference*, pp. 568–573, IEEE, 2010.

[16] A. Michalas, N. Komninos, and N. R. Prasad, "Mitigate dos and ddos attack in mobile ad hoc networks," *International Journal of Digital Crime and Forensics (IJDCF)*, vol. 3, no. 1, pp. 14–36, 2011.

[17] A. Michalas, N. Komninos, and N. Prasad, "Multiplayer game for ddos attacks resilience in ad hoc networks," in *Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*, pp. 1–5, Feb 2011.

[18] A. Michalas, N. Komninos, and N. R. Prasad, "Cryptographic puzzles and game theory against dos and ddos attacks in networks," *International Journal of Computer Research*, vol. 19, no. 1, p. 79, 2012.

[19] T. Dimitriou and A. Michalas, "Multi-party trust computation in decentralized environments," in *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, May 2012.

[20] T. Dimitriou and A. Michalas, "Multi-party trust computation in decentralized environments in the presence of malicious adversaries," *Ad Hoc Networks*, vol. 15, pp. 53–66, Apr. 2014.

[21] A. Michalas, V. A. Oleshchuk, N. Komninos, and N. R. Prasad, "Privacy-preserving scheme for mobile ad hoc networks," in *Computers and Communications (ISCC), 2011 IEEE Symposium on*, pp. 752–757, June 2011.

[22] A. Michalas and N. Komninos, "The lord of the sense: A privacy preserving reputation system for participatory sensing applications," in *Computers and Communication (ISCC), 2014 IEEE Symposium*, pp. 1–6, IEEE, 2014.

[23] A. Michalas, M. Bakopoulos, N. Komninos, and N. R. Prasad, "Secure amp; trusted communication in emergency situations," in *Sarnoff Symposium (SARNOFF), 2012 35th IEEE*, pp. 1–5, May 2012.