

9-1-2021

Reconsidering big data security and privacy in cloud and mobile cloud systems

Lo'ai A. Tawalbeh
Jordan University of Science and Technology

Gokay Saldamli
San Jose State University, gokay.saldamli@sjsu.edu

Follow this and additional works at: https://scholarworks.sjsu.edu/faculty_rsca

Recommended Citation

Lo'ai A. Tawalbeh and Gokay Saldamli. "Reconsidering big data security and privacy in cloud and mobile cloud systems" *Journal of King Saud University - Computer and Information Sciences* (2021): 810-819.
<https://doi.org/10.1016/j.jksuci.2019.05.007>

This Article is brought to you for free and open access by SJSU ScholarWorks. It has been accepted for inclusion in Faculty Research, Scholarly, and Creative Activity by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.



Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Reconsidering big data security and privacy in cloud and mobile cloud systems

Lo'ai A. Tawalbeh^{a,b,*}, Gokay Saldamli^{b,c}^a Computer Engineering Department, Jordan University of Science, Technology, Irbid 22110, Jordan^b Koc Lab, Department of Computer Science, University of California Santa Barbara, CA 93106, USA^c Computer Engineering Department, San Jose State University, San Jose, CA 95112, USA

ARTICLE INFO

Article history:

Received 7 March 2019

Revised 5 May 2019

Accepted 23 May 2019

Available online 29 May 2019

Keywords:

Cloud computing

Networked mobile cloud system

Big data security and privacy

ABSTRACT

Large scale distributed systems in particular cloud and mobile cloud deployments provide great services improving people's quality of life and organizational efficiency. In order to match the performance needs, cloud computing engages with the perils of peer-to-peer (P2P) computing and brings up the P2P cloud systems as an extension for federated cloud. Having a decentralized architecture built on independent nodes and resources without any specific central control and monitoring, these cloud deployments are able to handle resource provisioning at a very low cost. Hence, we see a vast amount of mobile applications and services that are ready to scale to billions of mobile devices painlessly. Among these, data driven applications are the most successful ones in terms of popularity or monetization. However, data rich applications expose other problems to consider including storage, big data processing and also the crucial task of protecting private or sensitive information.

In this work, first, we go through the existing layered cloud architectures and present a solution addressing the big data storage. Secondly, we explore the use of P2P Cloud System (P2PCS) for big data processing and analytics. Thirdly, we propose an efficient hybrid mobile cloud computing model based on cloudlets concept and we apply this model to health care systems as a case study. Then, the model is simulated using Mobile Cloud Computing Simulator (MCCSIM). According to the experimental power and delay results, the hybrid cloud model performs up to 75% better when compared to the traditional cloud models. Lastly, we enhance our proposals by presenting and analyzing security and privacy countermeasures against possible attacks.

© 2019 The Authors. Production and hosting by Elsevier B.V. on behalf of King Saud University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

The emerging technologies of cloud and mobile computing bring up the mobile cloud computing paradigm that creates great applications and services improving people's quality of life and organizational efficiency (Liu et al., 2015). Still increasing mobile device usage sparked the wide spread of products (e.g. social networks, education, healthcare, government, etc.) that some has

the capacity of producing enormous amount of data, sometimes also called big data. These technological advances are great but they reveal other problems that need to be addressed including storing and processing big data, protecting user privacy, and securing sensitive information (Yaqoob et al., 2016).

Big data consists of enormous amount of structured, semi-structured, and unstructured data that is generally tagged with implicit information and collected from different devices such as smartphones, personal computers, traffic cameras and sensors (Oussous, 2018). The term big here does only emphasize the enormous size of the data (mostly referred as terabytes, petabytes or zettabytes) but also describes various data types and data generation velocity (how frequent data is generated or collected). Big data processing can be initiated either upon request or in cycles based on the nature of the job to be completed. In general, this processing (sometimes also called big data analytics) involves to examine and analyze big data sets in order to make more informed decisions.

* Corresponding author at: Computer Engineering Department, Jordan University of Science and Technology, Irbid 22110, Jordan.

E-mail addresses: tawalbeh@just.edu.jo (L.A. Tawalbeh), gokay.saldamli@jsu.edu (G. Saldamli).

Peer review under responsibility of King Saud University.



Production and hosting by Elsevier

There are so many real-world challenges where big data analytics can be seen in action. For instance; in heavy industries, the machinery or equipment in the production lines should continuously be tracked and analyzed (sometimes every minute or even in shorter time intervals) so that any replacement and maintenance could be done timely (Wang et al., 2018). Such a task poses a real-world challenge where massive amounts of frequently collected operation data from thousands of equipment for multiple functionalities have to be analyzed mostly in real-time. Any failure in detecting a malfunctioning equipment in time could be extremely costly to a manufacturer since it could result in interrupting the whole production line.

Another example is the large patient record databases in hospitals. For various reasons, hospitals have to keep records of their patient's medical data safe and protected from unauthorized access (Kurdi, 2015). In general, this data should be highly available in case any kind of analysis to support a medical decision-making is needed. This could be a decision whether to perform a procedure or not as well as simply to determine whether a patient need to be discharged or readmitted. These decisions do not only back the patient's health but also reduce the cost (e.g. prevention of an unnecessary stay) and support a more sustainable healthcare system.

Governments on the other hand probably have the largest sets of data that has to be dealt with to patronize public on many levels (nations, states and cities). Others include sport teams that need to work with huge fan datasets to predict ticket sales and evaluate team strategies; and lastly, the social media that is playing an increasing role in our modern lives, changing the ways of marketing, advertising, manufacturing and many others. Having an enormous user base, social media companies have to process huge sets of data in order to give a better service to their end users and their business clients targeting these users.

For these and many other real-world problems, cloud infrastructures are considered as the appropriate medium that can process the floods of data (Ghasemi-Falavarjani et al., 2015). In here, process does not only mean data storage but also the analysis; most cloud providers offer data analytics services, sometimes also called Analytics as a Service (AaaS), built into the cloud infrastructure. AaaS offerings could be very rich facilitating the analytics on big and different types of data (Ardagna et al., 2017). With well-designed connection layers, cloud can accumulate all kinds of data from trusted origins and can organize them based on priorities. After the analysis, outcomes could be visualized in order to reveal the meaningful information requested and sent back to its destination. Moreover, cloud assures the necessary management and control tools underlining the (regulatory) governance guidelines while fulfilling the requirements of multi-national enterprises (Oussous, 2018).

There are three main known cloud architectures, namely: peer to peer (P2P), federated and centralized,

- **Centralization:** is more suitable for applications that require low communication delays and it is used in the computing clusters and datacenters of many cloud providers (Ferrer et al., 2019). By using this approach, clients are being tied to the closest data-center in order to avoid high communication latency because cloud resources are geographically partitioned over wide distances.
- **Federated cloud:** is used to construct large clouds by merging many smaller clouds together (Kahanwal and Singh, 2013). Federated architecture is beneficial when clients try to assure high level of confidentiality when distributing data geographically.
- **P2P cloud:** is based on extending the federated concept by building the cloud without using specific component for centralization and monitoring (Kumari et al., 2018). Its cloud

architecture consists of independent peers and resources, and the resource provisioning is conducted at a fairly low cost as a result of minimal management.

Fig. 1 shows a vertical representation of a typical P2P cloud architecture. In general, end users might not know which cloud architecture is in use. The only important thing to them is the quality of the service (QoS) they are receiving. In P2P clouds, QoS requirements and the terms of use should be outlined clearly in the Service Level Agreement (SLA) (Kumari et al., 2018).

On the other hand, among the most useful services provided by the P2P cloud systems is the reliable and secure data storage. In these cloud systems, data is divided into small blocks and redundantly distributed to multiple geographical locations that might be at different cities or even countries. This approach might help to prevent unauthorized users or hackers from accessing complete copy of these files.

In addition to hackers distraction, data decentralization would boost the speed of the as data access involves gathering the related portions of the data from different resources in parallel with better performance. This data storage technique can clearly ensure the content level protection so that reaching the stored data is almost impossible by those who have no access privileges. Moreover, storing data in such a way guarantees higher reliability because there would be many replicas of the same file but no single party would have a full control over any stored copy.

The power of big data analytics is mostly appreciated once data is analyzed and meaningful results are delivered to requesters. In general, cloud computing is considered as the most cost effective solution for big data analytics. Besides storing the data, the cloud computing environment provides the clients with several advanced tools in machine learning, artificial intelligence, and many others. Using these advanced techniques, users are able to explore and analyze all formats of data including videos, images, texts, etc. However, when dealing with such huge volumes of data, the traditional database approaches might not suit well. The database queries could get complicated and costly when many attributes have to be handled. On top of these scalability problems, there are security issues related to the data ownership and control. Storing and accessing data over clouds pose a big threat as data owners have to give the authority of data control and management to the cloud service providers.

In addition to the facts mentioned earlier, there are many motivations behind conducting this research. We are motivated by the huge amounts of big data being generated every second that need

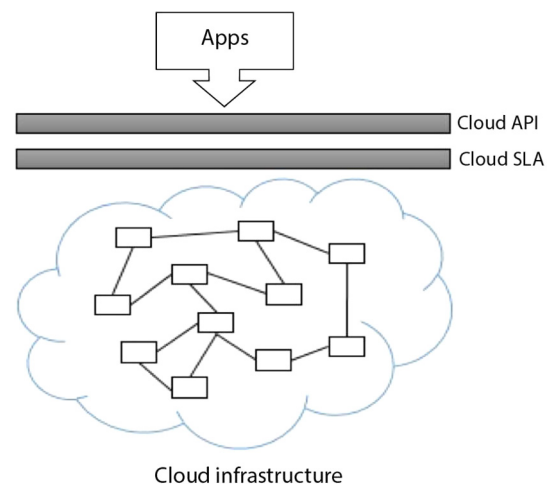


Fig. 1. Vertical representation for P2P cloud system.

efficient analysis and storage. As a real case example is the medical records in the healthcare sector where every patient has medical tests, results, X-rays and many other associated huge amounts of data that need processing and storage. Moreover, the user's data –in most cases– contain critical and essential information that need to be protected from cyber attacks which are becoming more sophisticated nowadays. As a real case example, recently there was a data breach on social media networks where millions of private records were attacked and revealed. Besides that, we read/hear about new advanced hacking techniques which didn't exist few years ago such as ransomware, spectrum, meltdown attack, and many others. Another important motivation is the need for a cloud/mobile cloud model that is both secure and efficient in order to meet the big data requirements.

Based on the above mentioned motivations, the contribution of this research can be summarized in the following points:

- Review the existing layered architectures of cloud systems and present an efficient solution for big data storage on the cloud environments.
- Explore the use of P2P Cloud Systems (P2PCS) for big data analytics.
- Investigate the security and privacy issues on cloud and mobile cloud systems. We explore relevant attacks and respective countermeasures to protect these systems from such possible attacks.
- Propose an efficient hybrid mobile cloud computing model based on cloudlets concept.
- Simulate the proposed model using Mobile Cloud Computing Simulator and obtain performance experimental results (delay and consumed power).

The next section revises the related literature. Section III presents the cloud architectures and P2P cloud/mobile cloud systems and how it suits the big data analytical needs. Section IV presents security considerations and attacks in cloud systems, and outlines possible countermeasures and protection mechanisms. In Section V we propose our Hybrid cloud model based on the cloudlet concept followed by the simulation results in Section VI. Conclusions and future work are presented in Section VII.

2. Related work

Recently, Peer to Peer Cloud Systems (P2PCS) has gained an increasing interest. There are some studies in the literature presenting several possible architectures to build cloud computing systems (see (Wang et al., 2018) and (Balasubramanian and Karmouch, 2017)). More recent work in (Kumari et al., 2018) focuses on building P2P-cloud system from reliable cloud resources with low cost to provide a wide range of services. The work in (Lo'ai et al., 2016) proposed mobile cloud computing model that can be used for a variety of useful applications. The proposed model can be used to store and analyze the data generated from different sensors (such as fire and motion sensors) and from IoT devices. The collected data will be sent to the mobile cloud model for analysis and making the appropriate decision efficiently in constrained environments.

On the other hand, there are many threats and possible attacks on cloud and mobile cloud systems that might compromise the privacy and the integrity of critical users' data (Mollah et al., 2017).

Other studies combine cloud infrastructures with specific services targeting particular industries. In other words, the cloud is built in a specific manner to serve specific services to the clients such as cloud computing for manufacturing or cloud computing for health care (Jemal et al., 2015).

The authors in (Lo'ai et al., 2016) discussed the useful integration of mobile cloud computing in the healthcare sector and proposed a mobile cloud system based on the cloudlet concept for healthcare applications. In the context of big data generation, the authors in (Booth et al., 2013) pointed out the common features to scientific data of all disciplines regardless of the source it was generated from. These common features include: massive scale; manipulated through large, distributed workflows, complexity, and accuracy.

In (AlDairi and Tawalbeh, 2018), the authors classified certain attacks on cloud and mobile cloud computing environments based on the strategies and countermeasures used to defend against these attacks. Also, they addressed cyber attacks on smart cities and their related embedded technologies. In (Gupta et al., 2015), the authors addressed security challenges and concerns associated with big data and mobile cloud computing. The authors in (Jhuria et al., 2013) addressed the issue of lack of security in cloud environments. They surveyed the traditional available encryption algorithms that are used to encrypt the stored data at the cloud environments.

3. Cloud architecture and big data

In this section, first we summarize the layered architecture of P2PCS as presented in (Kurdi, 2015) and propose a big data analytical model that we would fit to this architecture in sub-section 3-A. Next in sub-section 3-B, we discuss mobile cloud computing and big data.

3.1. P2P cloud systems

The P2P Cloud System contains a set of hosts or nodes (peers) that run identical processes (software components), that are organized and executed according to the layers of the provided architecture.

The first layer which is also called Peer Sampling Service (PSS) involves a simple gossip protocol (Jelasity et al., 2007) where each node gets a list of all neighbor nodes that it can speak to. Each neighboring node in the local view contains a form of ID (e.g., IP address) and a timestamp. Neighboring nodes go into the local view based on the time of the first interaction indicated by the timestamp.

Neighbors periodically share and merge their local views; removing the oldest entries from their local views to keep the list size fixed determined by the node itself. Since the list of nodes could possibly change after each message, the local view is a dynamic list. PSS is considered as a very efficient solution for decentralized environments where individual nodes have the responsibility of controlling the resources.

In the second layer, Slicing Service (SS), nodes are ranked according to the users' requests based on a certain criteria. When a user requests a specific allocation of nodes, all existing nodes that match the query will be grouped together to form a slice or a sub-cloud. For instance, a user can request the fastest 5% nodes to form a slice.

The third layer is called Aggregation Service (AS) in which cloud-wide parameters are provided to any node upon request without accessing the global cloud registry. Cloud-wide measurements include parameters that describe the status or the state of the cloud system such as the total number of nodes in the cloud, average load, utilization, etc. These values are generated using decentralized aggregation methods rather than a central unit. The data aggregated this way is consumed through a Monitoring Service (MS) via some APIs running on top of AS that also can be used to watch the states of the nodes and to display the network topology.

The T-Man protocol is used to connect the peers which belongs to the same slice in the P2PCS (Jelasity et al., 2009). T-Man is a gossip-based protocol that can construct any specific overlay network on top of an existing infrastructure including different topologies. Fig. 2 shows how the nodes that belong to the same slice are linked together with a ring overlay using T-Man protocol. The T-Man protocol creates subcloud and also manages the node failures by removing nodes that show specific failure from the existing overlay and re-link the remaining nodes together with a new overlay keeping the topology unchanged.

The P2PCS layered architecture also includes a Dispatcher which is responsible for translating the high level user commands to low level instructions that are sent to other peers. Another component in P2PCS is the Instance Management API (IMA) which allows the users to control resource creation, termination, and other operations. Other components in the P2PCS is the Storage system In P2PCS which is implemented separately as a distributed service. The authentication system in P2PCS is responsible for access control giving the necessary rights to authorized users.

We propose to use P2PCS as summarized above to store, organize, manipulate and move big data in the cloud. We think the combination of P2P and cloud computing is the best solution for big data management and analytics. On one hand, P2P networking facilitates the desired decentralization so that participants (peers) can keep their data under control and even share resources across clouds. On the other hand, cloud provides re- sources for storage, networking, and computing required for big data analytics that can be performed in parallel and with high flexibility.

The data storage system in (Kurdi, 2015) is implemented separately and the storage service is provided by set of operations including request data, allocate space, etc. A systematic approach for storing big data over P2P cloud is to split (partition) data into different sets according to access patterns (Liroz-Gistau et al., 2013). Fig. 3 shows how such a partitioning can be implement to hold and handle big data effectively and efficiently. In here, whenever a new data arrives, it is distributed to an appropriate partition based on a similarity metric computed over the existing partitions; larger partitions can be split further in time whenever necessary. Similarity metric could be based on different models; for instance, it could be measured by finding the number of queries access the data in one partition or the other.

3.2. Mobile cloud computing and big data

Big data innovation continues with advanced analytics that rests on cloud and mobile cloud computing (Fig. 4). There are more and more data-driven applications that make our lives easier in many aspects. Nevertheless, the value of the big data does not come from its huge volume, the real importance of this data comes from the ability to transform, refine and relate large amounts of data. This trend is also called data with intelligence, requires close collaboration between mobility and cloud computing (Liroz-Gistau et al., 2013). To reach this target, organizations would need to replace mobile applications with analytical mobile applications so that collected data would be filtered and analyzed beforehand.

For the mobility purpose, the analytical applications are built and hosted using the cloud computing technology and apps are

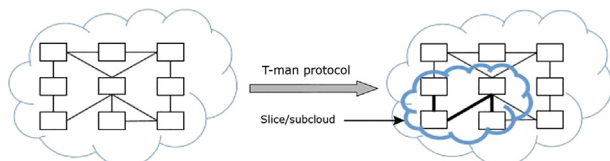


Fig. 2. Subcloud creation in P2PCS.

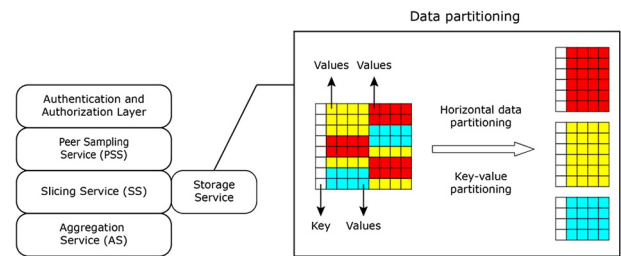


Fig. 3. Data partitioning in P2PCS storage.

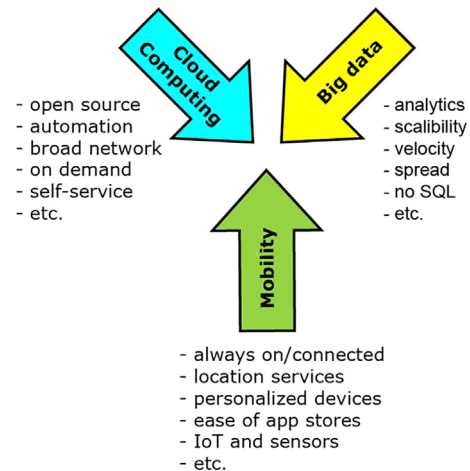


Fig. 4. Big data, cloud computing, mobility.

accessed via web servers that run on mobile browsers without considering the specifications of mobile operating system, memory and capacity.

The new service of providing analytics of complicated big data via mobile cloud computing to fulfil businesses needs by utilizing both Infrastructure as a service (IaaS) and Software as a Service (SaaS), is called Big Data as a Service (BDaaS). Security in this field occurs by ensuring security at both system and application layers (Gupta et al., 2015).

When it comes to application security, runtime applications that serve big data analytics on mobile devices have to be self-protected and self-aware applications. This means that existed firewalls and perimeters would not be sufficient anymore to provide the desired high-level security. After designing this kind of applications developers would need to employ adaptive access control to the applications. On the other side, at the system level, different technologies would be combined, such as machine learning and text mining to develop several programs for threats prediction and prevention. In the future, treating the security issue according to these two levels can help in ensuring advanced standards of protection against the dangerous threats of the modern digital world.

4. Security Considerations, attacks and countermeasure in Cloud/mobile cloud systems

4.1. Security considerations

Maintaining the data and applications private and integrated is one of the most important challenges in cloud computing environments. Many users have doubts of how much secure their data at the cloud and it will be under risk. These doubts are due to the lack of guaranties in an event of data loss or breach. From a legal perspective, these concerns could be settled through Service Level

Agreement (SLA) that clearly outline legal promises about privacy and data protection. Since security and privacy are very critical issues in cloud technology, in here, we investigate technological enhancements and practices that could address this challenge. After presenting most concerned attacks and threats that cloud computing suffer, we will explore the existing cryptographic methods that are considered to be the most appropriate in addressing these attacks (Gupta et al., 2015).

There are many known attacks on the cloud systems. We specifically present the attacks and threats that has big impact on the P2P cloud computing systems that include (Tawalbeh and Tawalbeh, 2017):

- **Data Breach:** happens when data that belongs to an entity accessed or captured in transaction by a third party without the data owner's permission. This could occur due to user errors, application vulnerabilities, or poor security settings but in order to protect P2PCS from such attacks, good security practices should be in place; such as hardware sharing must be limited and node accessing must be restricted to be exclusive to authorized parties. Implementing these and other countermeasures would prevent the attackers from deducing the P2PCS traffic by monitoring the usage of different P2PCS resources.

Among the countermeasures used to protect against certain attacks is to use the encryption. Although encryption does not directly solve all the above problems, it is still considered as one of the most powerful method for securing big data in the cloud and mobile cloud environments. Keeping data always encrypted would prevent attackers from deducing any meaningful information even if they have access to the cloud data storage.

4.2. Basic cryptographic methods

The most reasonable approach to big data protection is to design systems that respects the three key objectives of a security service, namely: confidentiality, integrity and availability. Confidentiality means to keep big data secret, so that no unauthorized entity would be able to reach, use or view data. The second service is the Integrity, from one side is about detecting any untrusted modification on big data, and from another side it is to make sure that result of any computation on big data is correct and consistent with the input. Availability is to make big data accessible and ready so that data owners would have no problems accessing their data whenever needed. These goals could be achieved by applying several cryptographic methods to the cloud system. In here, we summarize the basic methods and refer the reader to (Jhuria et al., 2013) and (Tawalbeh and Tawalbeh, 2017) for deeper analysis.

The most reasonable approach to big data protection is to design systems that respects the three key objectives of a security service: confidentiality, integrity and availability. Confidentiality is to keep big data secret, so that no unauthorized entity would be able to reach, use or view data. Integrity, from one side is about detecting any untrusted modification on big data, and from another side it is to make sure that result of any computation on big data is correct and consistent with the input. Availability is to make big data accessible and ready so that data owners would have no problems to access their data whenever needed. These goals could be achieved by applying several cryptographic methods to the cloud system. In here, we summarize the basic methods and refer the reader to (Jhuria et al., 2013) for deeper analysis.

The data stored in the cloud should be kept encrypted all the time so that any untrusted and un authorized identity should not be able to deduce any meaningful information even if they have access to the data storage. Basic cryptographic primitives include symmetric-key encryption where data owner need to have

a key that can be used for both encryption and decryption operations. In order to build a secure communication channel using symmetric-key methods, copies of this key have to be distributed before starting the encryption process. These algorithms are extremely fast and mostly used for block encryption in current cloud and mobile cloud deployments. Examples of the common symmetric encryption algorithms include (Tawalbeh and Tawalbeh, 2017):

- **3DES:** designed by cascading –now broken– DES (Data Encryption Standard) algorithm. 3DES has an effective key size of $3 \times 56 = 162$ -bits.
- **Blowfish:** designed in 1993 by Bruce Schneier. It is quite fast with variable key sizes 32 to 448 bits.
- **AES:** the new US standard algorithm used for bulk encryption with variable key lengths of 128,192 and 256. It is the fastest and has the smallest memory footprint among the algorithms in practice today.

The other category of encryption algorithms is the Asymmetric-key encryption that is mostly used to distribute keys of the symmetric-key systems and signing digital documents. These algorithms involve huge computations on large size operands. They might be slow but these are specialized secure algorithms. Examples include (Tawalbeh and Mohammad, 2010):

- **RSA:** is one of the first public-key cryptosystems based on the so called “factoring problem”. It is introduced in 1978 for the first time and still the most widely used system in practice. Size of the private-key is considered to be equivalent to the size of the modulus (which is the number to factorize). Current applications need modulus size larger than 2048 bits, applications needs high security can go up to 16,386 bits of key sizes.
- **Diffie-Hellman (DH) key exchange:** is one of the first public-key protocols used for securely exchanging cryptographic keys over a non-secure communication channel. DH is based on the discrete logarithm problem over the multiplicative group of integers modulo n (a.k.a DH group). To have a secure DH key exchange, DH group should be at least 2048 bits.
- **Elliptic-curve cryptography (ECC):** is technique that can be used for encryption, digital signatures and key exchange. Its security is based on complexity of the discrete logarithm problem on the elliptic curve group over finite fields. ECC requires smaller keys compared to other public-key systems; NIST recommends 15 ECC parameters having key sizes between 163 and 571 bits.

The Asymmetric encryption algorithms provides high level of security but they require efficient implementations of the underlying finite fields modular mathematical operations (Tawalbeh et al., 2012); (Lo'ai et al., 2004).

On the other hand, Hash functions are the methods of creating message digests (hash values) that are used for checking digital integrity needed in many places including in signing digital documents. These hashing algorithms can be implemented efficiently in software and hardware (Moh'd et al., 2010). Example of main known secure hash functions are SHA-2 and SHA 3.

- **SHA-2 family:** is a set of cryptographic hash functions introduced by the National Security Agency (NSA). SHA-2 is based on the design of MD5 and SHA-1 algorithms (both got recently broken). The SHA-2 family (i.e. SHA-224, SHA-256, SHA-384, SHA- 512, SHA-512/224, SHA-512/256) consists of six hash functions with digests that are 224, 256, 384 or 512 bits.
- **SHA-3:** is the most recent (August 5, 2015) Secure Hash Algorithm that is selected through a public competition like as AES. SHA-3 can have variable length digest size but it has settings providing fixed sizes of 224, 256, 384 or 512 bits.

4.3. Advanced cryptographic methods

There are advanced encryption techniques that have various features in handling big data such as functional encryption, identity based encryption, attribute based encryption, homomorphic encryption (HE), verifiable computation (VC) and secure multiparty computation (MPC). Apart from these public-key methods, there is Format Preserving Encryption (FPE) based on symmetric key encryption particularly conserve the length or the format (keep numbers as numbers, or characters sets) of the original messages. A similar idea is Format Preserving Hashing (FPH) performing secure hashing while keeping the format of the plaintext.

Format Preserving Encryption: FPE encrypts a plaintext of some specified format into a ciphertext of the same format. For example, encrypting a social-security number into a social-security number as seen in Fig. 5. Existing FPE schemes are built with Feistel networks using block ciphers. In 2016, the National Institute of Standards and technology (NIST) released an FPE standard (Dworkin, 2016).

FPE particularly attract businesses having legacy soft-ware and willing to protect their data in compliance with national standards such as PCI DSS and HIPAA as well as the European Commission's General Data Protection Regulation (GDPR).

FPE is extremely fast as it is based on block ciphers. It can use any type of encryption including AES. Therefore, it quite suitable for encrypting big data and provide confidentiality but its analytics capabilities are limited; even very simple queries might need re-encryption. Nevertheless, certain simple functions that does not need aggregation (these need homomorphic encryption) or comparisons could be carried with FPE protected big data.

These functions could be improved with some smart partitioning or tagging with extra meta-data. For example, a simple sql search “select * $E_k(\text{value})$ (i.e. ciphertext)” or finding the most frequent value in a given column could be handled without needing an FPE re-encryption. However, finding values larger or smaller than a threshold could not be achieved clear-cut. One has to design some custom data partitioning (see Fig. 3) not only based on simple key-value fashion but more fine-grained (see Fig. 5) and maximized separation using multiple layers of FPE. To summarize, FPE is practical and provides decent solutions for clouds not very complicated requirements.

Advanced public-key techniques: Asymmetric methods offer great features; for instance, with functional encryption a user having a secret key would be able to learn the input of a function of what the ciphertext is encrypting. Although these methods mostly suffer from complexity of the intensive computations they involve, some of them still have niche deployments.


 <div style="display: flex; justify-content: space-between;"> <div> <p>Tax ID</p> <p>934-72-1615</p> </div> <div> <p>First name: James</p> <p>Last name: Smithson</p> <p>SSN: 934-72-1615</p> <p>DOB: 01-17-1949</p> </div> </div>		
FPE mode	461-16-1615	First name: Qrpcf Last name: Uerrthwk SSN: 461-16-1615 DOB: 11-12-1973
Regular AES	Rã4d~†çflyx»µs ømÛEmäΩ;»	÷â;µvÄ'ôÄMl±ëWJ2\Xµä'A ¥û,îÄcñ∞ 223XnSçÄö _fY &Z¶'ad231m,,rwrwwXnS

Fig. 5. FPE's fine-grained encryption protects selected (sub) fields in a database; red text shows sensitive information that is encrypted.

The most appropriate public-key techniques for our purposes are homomorphic encryption (HE), verifiable computation (VC) and secure multiparty computation (MPC). In homomorphic encryption, computations occur after encrypting big data and keep them in the cloud. In other words any computation will only take place on the ciphertext version of data and not on the plaintext.

Cryptosystems that let specific operations to be performed on the resulted ciphertexts are called partially homomorphic cryptosystems such as unpadding RSA and ElGamal cryptosystem, whereas cryptosystems that allow arbitrary computations on the resulted ciphertext are called fully homomorphic schemes such as Gentry's cryptosystem (Gentry, 2009).

The fully homomorphic schemes are more capable than partially homomorphic schemes in ensuring confidentiality for cloud computing environments, and every analytical operation on big data have to be performed only on the encrypted stored version (see Fig. 6). As many advanced cryptosystem, fully homomorphic scheme offers great functionality in theory but this scheme is considered very inefficient because of its poor running time. Additionally, it requires all recipients to share one encryption key even if they belong to different ends, which makes the scheme unmanageable. Moreover, it does not allow sharing more than a key for encryption and this makes it less appropriate for P2P cloud environments in addition to ensuring only confidentiality across the cloud.

The other advanced non-traditional crypto technique is the Verifiable Computation (VC). In VC specific operations on big data are outsourced to a third party called prover (receiver) who takes big data or part of it from the data owners then performs requested operations, and returns the output along with a proof of results correctness.

We should note that in this scheme, integrity is highly achieved while confidentiality is not but the good thing about verifiable computation is the owner verifies the proof to make sure that the result is accurate. However, the process of proof verification has to be easier and cost much less than doing the actual computation, otherwise it would be more efficient for the owners to perform the operation themselves as can be seen from Fig. 7.

Therefore, the efficiency of this scheme depends on two measures; first, the running time of proof verification at the owner side and secondly, the time needed to construct a proof at the receiver side. To give some benchmarks; the best registered running time

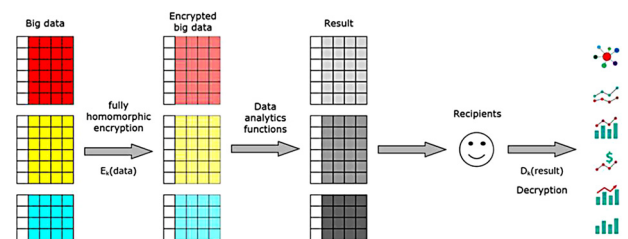


Fig. 6. Homomorphic encryption for protecting big data.

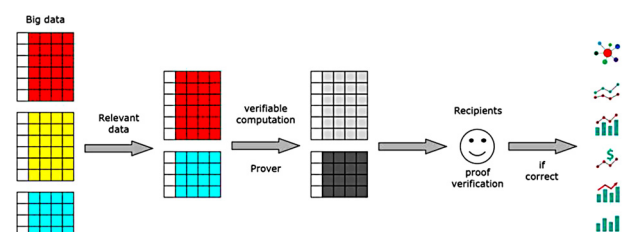


Fig. 7. Verifiable computation for protecting big data.

for proof verification reported in (Parno et al., 2013) 10 ms. On the other hand, the time for constructing a proof takes much longer time: for example, it is reported 31 and 144.4 s in (Ben-Sasson et al., 2014) and (Parno et al., 2013) respectively. With these performance figures, verifiable computation still does not reach the required efficiency needed for big data analytics.

Another advanced public key technique to secure big data in cloud is the secure multi-party computation (MPC). In this scheme, multiple data owners perform the same function on their data at the same time while keeping the data undercover. In other words, no owner knows anything about others data and all owners care only about the output of the function and about their data. (see Fig. 8).

Multi-party secure computation (Ben-Or et al., 1988) ensures both integrity and confidentiality and it is considered the most efficient scheme among the three public-key schemes we mentioned in earlier. The practical MPC systems and their applicability to cloud computing systems make it a good choice be employed to ensure security when treating big data over P2P clouds. However, the ensured confidentiality can be broken easily when an adversary corrupts enough number of participants who are participating in the same computation and reaches their secret data.

Since not all encryption methods are efficient, people go to hardware solutions to support the needs of their cloud system. Nowadays, most high-end microprocessor comes with hardware crypto support such as AES and some secure hash function but not any public-key algorithms. However, in most case these are not enough for the needs of cloud providers if they want to deploy some proven encryption method.

4.4. Secure Crypto-Processor

Secure crypto-processor, secure co-processor and crypto co-processor are all names for one secondary and specific microprocessor that is dedicated to provide high throughput crypto operations to its master framework. The main functionality of a secure crypto-processor should be designed after outlining possible threats and adversaries that can disturb the workflow of any system. Co-processor would support and perform the computations of any chosen security solution. Therefore, the importance of co-processors comes from being dedicated for only security issues while maintaining the performance of systems they belong to.

Hardware Security Module (HSM) that most cloud providers offer as a part of their key management service (Luo et al., 2018) is an example of a crypto co-processor. HSMs provide secure key storage and cryptographic operations within a tamper-resistant hardware. Hence, these devices are capable of detecting and protecting any tampering over data (mostly the encryption keys). In such cases, secure crypto-processor clears the system's memory as well as its internal RAM holding any sensitive information.

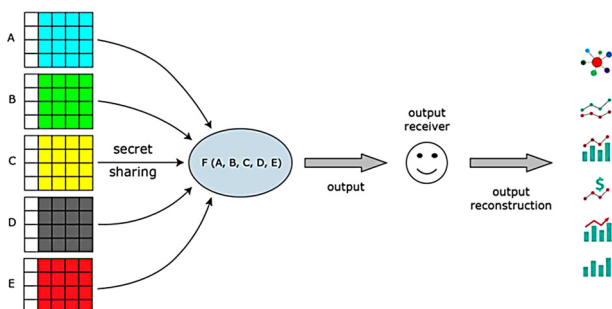


Fig. 8. Secure multi-party computation for protecting big data.

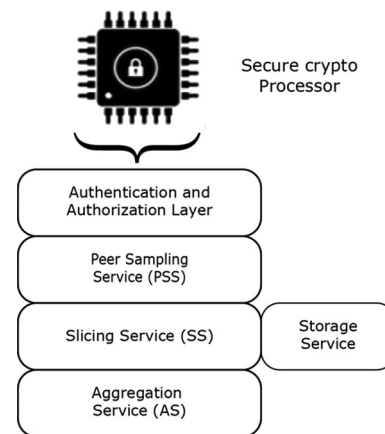


Fig. 9. A secure crypto-processor supporting P2PCS architecture.

Secure Crypto-Processor can play efficacious role in assuring security while dealing with big data analytics. They can be embedded to the cloud infrastructure and be responsible about all encryption and decryption operations on big data. In case of P2PCS, a co-processor such as seen in Fig. 9 can be deployed to each computer node in the network, so all of the encryption and decryption operations would be separated from other security operations. One challenge will be the efficiency that is totally determined by how efficiency of the hardware implementation of the finite field arithmetic operations that is embedded in the crypto-processor (Tenca and Tawalbeh, 2004).

5. The proposed hybrid cloud/Mobile cloud model

To prove the concept, we propose a hybrid mobile cloud computing model based on cloudlet concept.

There are different mobile cloud computing models. Among these models is the cooperative cloudlet model (Bahwairath and Tawalbeh, 2016). In the cooperative model, all cloudlets are connected to the enterprise cloud and this cloud is used as a centralized powerful infrastructure of cooperative cloudlets that usually work at intermediate levels and cooperate with each other to perform requested services for mobile users. The nearest cloudlet (e.g. CL1) to the user should be able to fulfill the task requested by that user. If that service is not available in CL1, then it should forward that task request to the next nearest cloudlets (e.g. CL2), and so on. The result will be returned back to the user in the same route it was forwarded. If not of the cloudlets are able to execute task, then it will be sent to the enterprise cloud (Bahwairath and Tawalbeh, 2016).

The drawbacks of the cooperative model is that at the worst case scenario the task has to pass through all the cloudlets in the model before it got executed. In the best case scenario the task will be executed at the first cloudlet. On average, if there is N cloudlets in the model, then the task will bounce through N/2 cloudlets encountering extra delay and more power consumption.

Another model is the centralized cloudlet model where are the cloudlets are connected to a master cloudlet that manages the task distribution among them. This approach suffers from the extra latency needed to send the task first to the master cloudlet then being distributed to the appropriate cloudlet in the model.

Driven by this motivation, we proposed a Hybrid cloudlet model that combines the concepts of centralized and cooperative approaches. And since we are studying the big data analysis using cloud computing, we considered the healthcare section as a case study. Fig. 10 shows the proposed hybrid model.

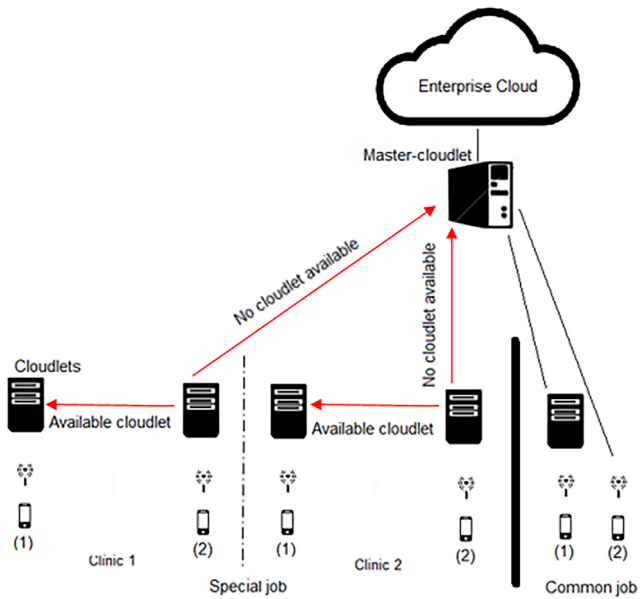


Fig. 10. Hybrid mobile cloud computing model.

In the proposed hybrid model, we apply the concepts of both the cooperative and the centralized models. The cooperative feature in our model is implemented inside each department (clinic) in the hospital. All the cloudlets in one department work cooperatively to serve the departments special jobs. When we compare between the worst case in the cooperative model and the worst case in the hybrid model, we find that a special job in the Hybrid model will be transferred through less number of cloudlets to be executed compared to the number of cloudlets that job will be transferred through in the cooperative model.

This is due to the fact that the number of distributed cloudlets in one department (hybrid model) is obviously less than the number of cloudlets in the whole hospital (cooperative model) where the task will pass through all these cloudlets in the hospital before it is executed while it only will be transferred within the cloudlets of that department in our hybrid model.

Also, the centralized feature is applied to our model since we are using the concept of one master-cloudlet. But we modified the tasks of the master cloudlets to include serving all common jobs that coming from all the hospitals departments (not special job for a specific clinic), while it was only responsible for routing the requests to other cloudlets in the original centralized approach.

6. Simulation results

There are many simulation tools for cloud environments. The work in (Bahwairath et al., 2016) compares the most common simulators and presents their advantages and disadvantages. Our proposed Hybrid cloud/mobile cloud model is simulated using Mobile Cloud Computing Simulator (MCCSIM) which is designed mainly for mobile cloud environments based on the CloudSim simulator (Calheiros et al., 2010). Fig. 11 shows the MCCSIM interface.

There is no doubt about the high level of security that the non-traditional encryption algorithms will provide for the cloud environment. But in order to measure the performance of the suggested non-tradition encryption techniques discussed in the previous section, we need to characterize it as a task that will be executed in both mobile cloud computing models (cooperative

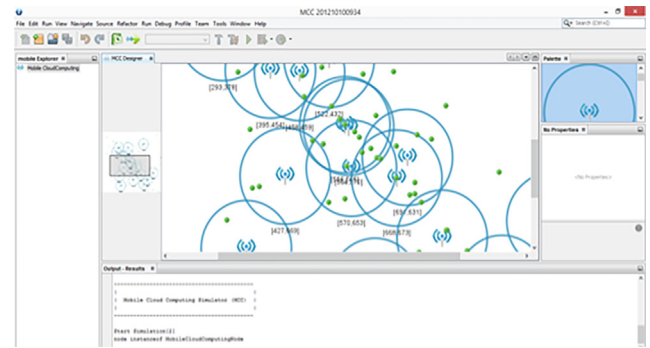


Fig. 11. MCCSIM Interface.

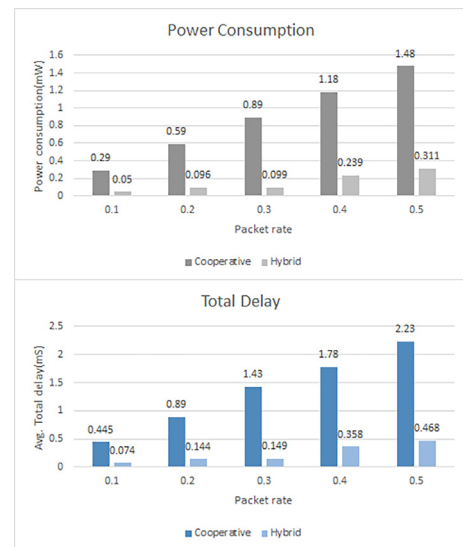


Fig. 12. Cooperative vs. Hybrid Mobile Cloud Models Comparison.

and hybrid) and compare their performance in terms of time delay and power consumption.

Using the MCCSIM, the task is measured by the packets size per time unit (packet rate) which will be identical in both mobile cloud models (in the range from 0.1 to 0.5). Fig. 12 shows the simulation delay and consumed power results when applying the same task (represents the encryption) on the cooperative model and our proposed hybrid model.

Fig. 12 shows that on average our proposed Hybrid model consumes about 75% less power than the cooperative model at different packet rates. Moreover, our model has less delay than the cooperative model by up to 80% at certain packet rates.

In the cooperative model (Bahwairath and Tawalbeh, 2016), the task is transferred to the next nearest cloudlet when the first contacted cloudlet cannot perform it. If it is found at the next cloudlet, it will be executed and returned back to the first one. If not, it will be forwarded to the third cloudlet and so on till the task is executed. This routing procedure results in longer path causing the extra delay and power consumption.

7. Conclusions and future work

There are many computing trends providing useful services to enhance the individuals lives and increased the organizations effi-

ciency. These trends include cloud and mobile cloud computing but along with these technologies there are many associated challenges that should be taken in consideration such as users privacy and data security. In this research we studied the recent emerging technologies that include: cloud systems, mobile cloud computing, P2P cloud systems, big data and storage solutions. Also, we addressed security and privacy issues associated with these technologies, and presented the important attacks that have major influence on cloud computing systems with the traditional existing countermeasures used to defeat these attacks.

Moreover, we studied the layered P2PCS architecture and its importance in big data analysis. Then we studied the possibility of applying new countermeasures against security threats. In particular, we presented four non-traditional encryption techniques and analyzed the visibility of using them in terms of performance parameters to secure big data in cloud environments, namely, format preserving encryption, homomorphic encryption, verifiable computation, and secure multi-party computations.

Adding to that, we proposed hybrid mobile cloud model and conducted simulation to prove the concept of using mobile cloud computing models in real life big data application (healthcare case). We also measured and compared the performance parameters (delay and power consumption) for our model and previously proposed cooperative mobile cloud model in the literature.

Finally, we conclude that the cloud and mobile cloud computing environments are suitable to host and analyze big data. There are many developed and developing security attacks that threaten the cloud and mobile cloud computing environments. Protecting big data at such environments needs new efficient countermeasures. The next step would be to investigate the real-time deployment of modern cryptographic methods such as the homomorphic encryption and Format Preserving Encryption in real world cloud environments to secure big data.

Funding Source Declaration: This work was supported as a sabbatical leave grant to Dr Lo'ai Tawalbeh from the deanship of research at Jordan University of Science and Technology.

Declaration of Competing Interest

There is no conflict of interest related to this submitted manuscript

Acknowledgments

This research was supported financially by Jordan University of Science and Technology (Jordan)-Sabbatical leave reward. And it was conducted at Um Al-Qura University (KSA) and University of California-Santa Barbara, USA.

References

- Aldairi, A., Tawalbeh, L., 1086. Cyber security attacks on smart cities and associated mobile technologies. *Proc. Comput. Sci.* 109, 1086–1091.
- Ardagna, C.A., Bellandi, V., Ceravolo, P., Damiani, E., Bezzi, M., Hebert, C., 2017. A model-driven methodology for big data analytics-as-a-service. In: 2017 IEEE International Congress on Big Data (BigData Congress), pp. 10–112.
- Bahwairath, Khadijah, Benkhelifa, Elhadi, Jararweh, Yaser, Tawalbeh, Mohammad A., 2016. Experimental comparison of simulation tools for efficient cloud and mobile cloud computing applications. *EURASIP J. Inf. Security* 2016 (1), 15.
- Bahwairath, K., Tawalbeh, L., 2016. Cooperative models in cloud and mobile cloud computing. In: 23rd International Conference on Telecommunications (ICT). IEEE, pp. 1–4.
- Balasubramanian, V., Karmouch, A., 2017. An infrastructure as a service for mobile ad-hoc cloud. In: 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), pp. 1–7.
- Ben-Or, M., Goldwasser, S., Wigderson, A., 1988. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, ser. STOC '88. ACM, New York, NY, pp. 1–10.
- Ben-Sasson, E., Chiesa, A., Tromer, E., Virza, M., 2014. Succinct non-interactive zero knowledge for a von Neumann architecture. In: 23rd USENIX Security Symposium (USENIX Security 14). USENIX Association, San Diego, CA, pp. 781–796.
- Booth, G., Soknacki, A., Somayaji, A., 2013. Cloud security: attacks and current defenses Albany, NY. In: 8th Annual Symposium on Information Assurance (ASIA'13), pp. 56–62.
- Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F., Buyya, R., 2010. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience* 41 (1), 23–50.
- M. Dworkin, "Recommendation for block cipher modes of operation: Methods for format preserving encryption," NIST.SP.800-38G, Mar. 2016.
- Ferrer, A.J., Marques, J.M., Jorba, J., 2019. Towards the decentralized cloud. *ACM Comput. Surv.* 51 (6), 1–36.
- Gentry, C., 2009. A Fully Homomorphic Encryption Scheme (Ph.D. dissertation). Stanford University, Stanford, California.
- Ghasemi-Falavarjani, S., Nematbakhsh, M., Ghahfarokhi, B.S., 2015. Context-aware multi-objective resource allocation in mobile cloud. *Comput. Electr. Eng.* 44, 218–240.
- Gupta, D., Chakraborty, P.S., Rajput, P., 2015. Cloud security using encryption techniques. *Int. J. Adv. Res. Comput. Sci. Softw.* 5, 425–429.
- Jelasity, M., Voulgaris, S., Guerraoui, R., Kermarrec, A.-M., van Steen, M., 2007. Gossip-based peer sampling. *ACM Trans. Comput. Syst.* 25 (3).
- Jelasity, M., Montresor, A., Babaoglu, O., 2009. T-man: gossip-based fast overlay topology construction. *Comput. Netw.* 53 (13), 2321–2339.
- Jemal, H., Kechaou, Z., Ayed, M.B., Alimi, A.M., 2015. Mobile cloud computing in healthcare system. In: Núñez, M., Nguyen, N.T., Camacho, D., Trawinski, B. (Eds.), *Computational Collective Intelligence*. Springer International Publishing, Cham, pp. 408–417.
- Jhuria, M., Singh, S., Nigoti, R., 2013. A survey of cryptographic algorithms for cloud computing. *Int. J. Emerg. Technol. Comput. Appl. Sci.* 05, 141–146.
- Kahanwal, B., Singh, T.P., 2013. The distributed computing paradigms: P2P, grid, cluster, cloud, and jungle. *CoRR*.
- Kumari, Priti, Kaur, Parmmeet, 2018. A survey of fault tolerance in cloud computing. *J. King Saud Univ.-Comput. Inf. Sci.*
- Kurdi, Heba A., 2015. HonestPeer: An enhanced EigenTrust algorithm for reputation management in P2P systems. *J. King Saud Univ.-Comput. Inf. Sci.* 27 (3), 315–322.
- Liroz-Gistau, M., Akbarinia, R., Pacitti, E., Porto, F., Valduriez, P., 2013. Dynamic workload-based partitioning algorithms for continuously growing databases. *Trans. Large-Scale Data- Knowledge-Centered Syst.* 12, 105–128.
- Liu, J., Ahmed, E., Shiraz, M., Gani, A., Buyya, R., Qureshi, A., 2015. Application partitioning algorithms in mobile cloud computing: taxonomy, review and future directions. *J. Netw. Comput. Appl.* 48, 99–117.
- Lo'ai, A.T., Bakhader, W., Mehmood, R., Song, H., 2016. Cloudlet-based mobile cloud computing for healthcare applications. In: 2016 IEEE Global Communications Conference (GLOBECOM). IEEE, pp. 1–6.
- Lo'ai, A. Tawalbeh, Tenca, Alexandre F., 2004. An algorithm and hardware architecture for integrated modular division and multiplication in GF (p) and GF (2n). In: *Proceedings of the Application-Specific Systems, Architectures and Processors*, 15th IEEE International Conference, pp. 247–257.
- Lo'ai, A. Tawalbeh, Bakhader, Waseem, 2016. A mobile cloud system for different useful applications. In: *Future Internet of Things and Cloud Workshops (FiCloudW)*, IEEE International Conference on. IEEE, pp. 295–298.
- Luo, S., Hua, Z., Xia, Y., 2018. TZ-KMS: a secure key management service for joint cloud computing with ARM TrustZone. In: *IEEE Symposium on Service-Oriented System Engineering (SOSE)*, pp. 180–185.
- Mohd, Abidalrahman, Aslam, Nauman, Marzi, Hosein, Tawalbeh, L.A., 2010. Hardware implementations of secure hashing functions on FPGAs for WSNs. *Proceedings of the 3rd International Conference on the Applications of Digital Information and Web Technologies (ICADIWT)*.
- Mollah, M.B., Azad, M.A.K., Vasilakos, A., 2017. Security and privacy challenges in mobile cloud computing: survey and way ahead. *J. Netw. Comput. Appl.* 84, 38–54.
- Oussous, Ahmed, Benjelloun, Fatima-Zahra, Lahcen, Ayoub Ait, Belfkih, Samir, 2018. Big data technologies: a survey. *J. King Saud Univ.-Comput. Inf. Sci.* 30 (4), 431–448.
- Parno, B., Howell, J., Gentry, C., Raykova, M., 2013. Pinocchio: Nearly practical verifiable computation. In: *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, ser. SP '13. IEEE Computer Society, Washington, DC, USA, pp. 238–252.
- Tawalbeh, Lo'ai, Jararweh, Yaser, Mohammad, Abidalrahman, 2012. An integrated radix-4 modular divider/multiplier hardware architecture for cryptographic applications. *Int. Arab J. Inf. Technol.* 9 (3).
- Tawalbeh, L.A., Mohammad, Abidalrahman, Gutub, Adnan Abdul-Aziz, 2010. Efficient FPGA implementation of a programmable architecture for GF (p) elliptic curve crypto computations. *J. Signal Process. Syst.* 59 (3), 233–244.

- Tawalbeh, L.A., Tawalbeh, H., 2017. Lightweight crypto and security. In: Song, H., Fink, G.A., Jeschke, S. (Eds.), *Security and Privacy in Cyber-Physical Systems: Foundations, Principles, and Applications*. Wiley, pp. 243–261.
- Tenca, Alexandre F., Tawalbeh, L.A., 2004. Algorithm for unified modular division in $GF(p)$ and $GF(2^n)$ suitable for cryptographic hardware. *Electron. Lett.* 40 (5), 304–306.
- Wang, J., Zhang, W., Shi, Y., Duan, S., Liu, S., Industrial big data analytics: Challenges, methodologies, and applications, <https://arxiv.org/pdf/1807.01016.pdf>, April 2018.
- Yaqoob, I., Ahmed, E., Gani, A., Mokhtar, S., Imran, M., Guizani, S., 2016. Mobile ad hoc cloud: a survey. *Wireless Commun. Mobile Comput.* 16 (16), 2572–2589.