EasyChair Preprint

№ 7460

A Fog Computing Solution for Advanced Security, Storage Techniques for Platform Infrastructure

Anil Kumar Gardasu and Rohith Kumar Kotha

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

February 14, 2022

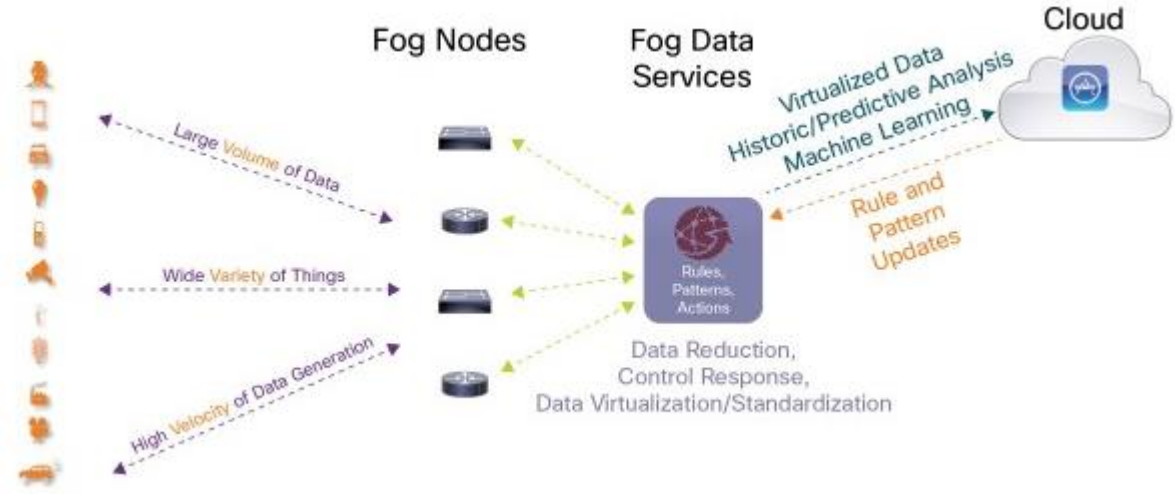
(1) Title of the thesis:

A Fog Computing Solution For Advanced Security, Storage Techniques for Platform Infrastructure

(2) Introduction:

Fog computing is defined as a distributed computing paradigm that fundamentally extends the services provided by the cloud to the edge of the network. Cisco defines it as fog computing is considered as an extension of the cloud computing paradigm from the core of network to the edge of the network. It facilitates the computation, storage and networking between the end devices and the traditional cloud servers. Instead of running application only in the cloud, fog computing involves the cloud as well as the edge devices between the end devices and cloud servers to run the application. Fog computing takes advantages of both edge and cloud computing while it benefits from edge devices' close proximity to the endpoints, it also leverages the on-demand scalability of cloud resources[8]. It basically reduces the load on the cloud server by efficiently using the resources available in the edge nodes to do partial computation and also it reduces the traffic to the cloud server by doing filtering operations in the nodes. There are mainly two concepts which are usually confused with fog computing. These concepts are Mobile Edge Computing (MEC) and Mobile Cloud Computing (MCC). MCC basically suggests that both the data storage and data processing is done outside the mobile in a cloud. So it moves the data and computing power

from the individual mobile to the cloud. MEC is similar to that of a Cloudlet.



Architecture of fog computing

ISSUES Fog computing extends cloud computing and acts on Internet of Things. These devices, called the fog nodes can be deployed in any environment with a network connection. Fog computing has additional storage resources at the edges to process the requirements. Hence, the Fog server needs to adapt its services leading to management and maintenance cost. In addition, the operator needs to encounter the following issues: Privacy Fog computing being dominated by wireless primarily, there is a big concern for network privacy. Network operator generates configurations manually, fog nodes being deployed at the edge of Internet, massive maintenance cost is involved. The leakage of private data is gaining attention while using networks. The end users are more accessible to the Fog nodes. Because of this, more sensitive information is collected by Fog nodes than remote cloud. Encryption methods like HAN (Home-Area Network) can be used to counter these issues. Security The main security issue is the authentication of the devices involved in fog computing at different gateways. Each appliance has its own IP address. A malicious user may use a fake IP address to access information stored on the particular fog node. To overcome this access control an intrusion detection system has to be applied at all layers of the

platform Network Management Being connected to heterogeneous devices, managing the fog nodes, the network, connection between each nodes will be burden unless SDN and NFV techniques are applied .Placement of Fog Servers Placing a group of fog servers in such a way that they deliver maximum service to the local requirements is an issue. Analyzing the work done in each node in the server before placing them reduces the maintenance cost. Delay in Computing Delays due to Data aggregation, Resource over-usage reduces the effectives of services provided by the fog servers, causing delay in computing data. Data Aggregation should take place before data processing, Resource-limited fog nodes should be designed scheduling by using priority and mobility model.

Difference between cloud computing and fog computing

Cloud computing technology provides various types of services that are categorized into three groups:

- **IaaS (Infrastructure as a Service)** — a remote data center with resources such as data storage capacity, processing power and networking.
- **PaaS (Platform as a Service)** — a development platform with tools and components for creating, testing and launching applications.
- **SaaS (Software as a Service)** — ready-made software tailored to a variety of business needs.

Connecting your company to the cloud, you get access to the above-mentioned services from any location and via different devices. Hence, availability is the greatest advantage. Moreover, there is no need to maintain local servers and worry about downtimes — the vendor supports everything for you, saving you money.

The integration of the Internet of Things with the cloud is a cost-effective way to do business. Off-premise services provide the necessary scalability and flexibility to manage and analyze data gathered by connected devices, while specialized platforms (e.g. Azure IoT Suite, IBM Watson, AWS, Google Cloud IoT) give developers the power to create IoT apps without big investments into hardware and software.

Since connected devices have limited storage capacity and processing power, the integration with cloud computing comes to assistance:

- **Improved performance** (the communication between IoT sensors and data processing systems is faster)
- **Storage capacities** (highly scalable and unlimited storage space are able to integrate, aggregate and share the enormous amount of data)
- **Processing capabilities** (remote data centers provide unlimited virtual processing capabilities on-demand)
- **Reduced costs** (license fees are lower than the cost of the on-premise equipment and its continuous maintenance)

Cons of Cloud for IoT

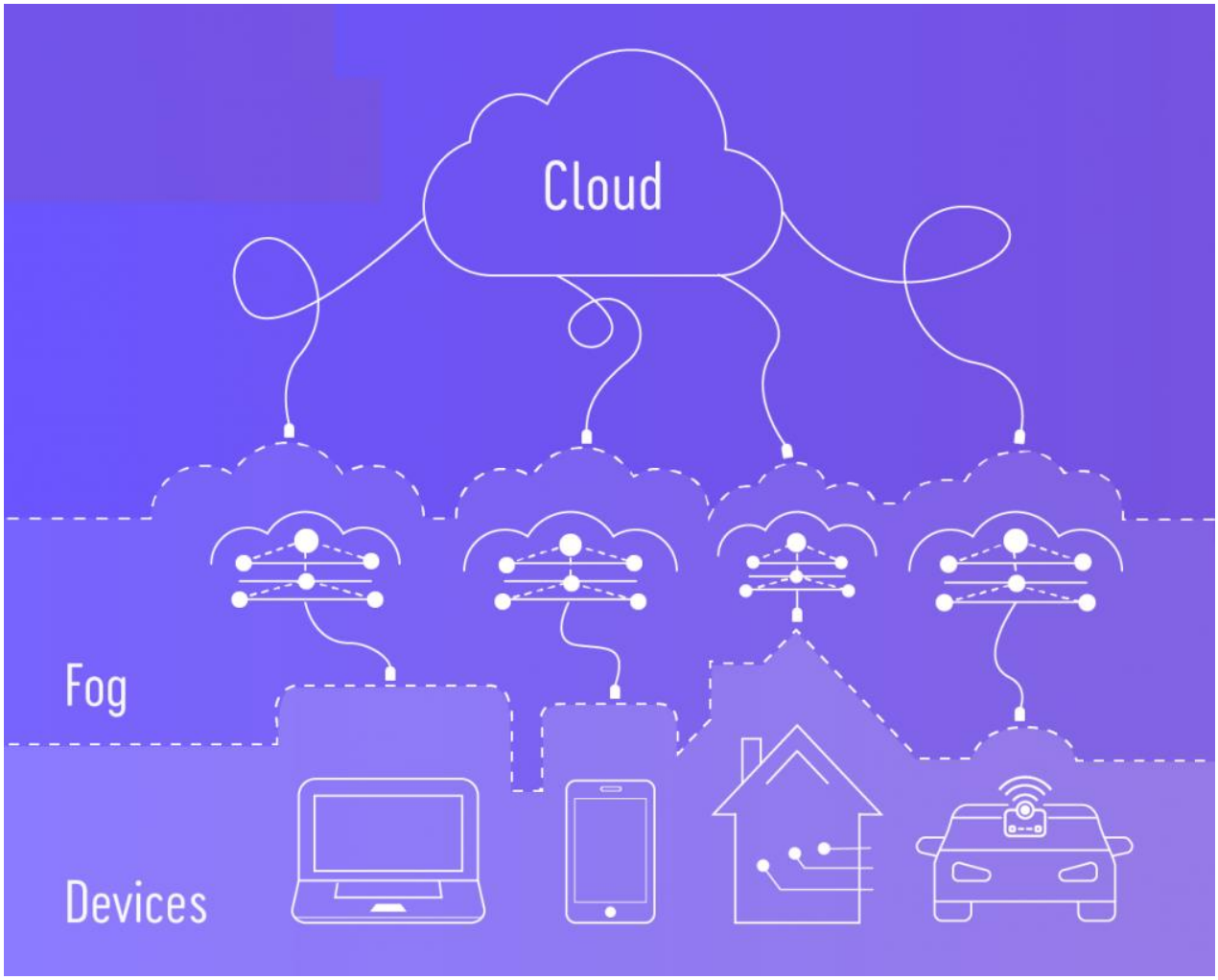
Unfortunately, there is nothing immaculate, and cloud technology has some downsides, especially for the Internet of Things services.

- **High latency** (more and more IoT apps require very low latency, but cloud can't guarantee it because of the distance between client devices and data processing centers)
- **Downtime** (technical issues and interruptions in networks may occur for any reason in any Internet-based system and make customers suffer from an outage; many companies use multiple connection channels with automated failover to avoid problems)
- **Security and privacy** (your private data is transferred through globally connected channels alongside thousands of gigabytes of other users' information; no surprise that the system is vulnerable to cyberattacks or data loss; the problem can be partially solved with the help of hybrid or private clouds)

Fog Computing:

The term *fog computing (or fogging)* was coined by Cisco in 2014, so it is new for the general public. Fog and cloud computing are interconnected. In nature, fog is closer to the earth than clouds; in the technological world, it is just the same, fog is closer to end-users, bringing cloud capabilities down to the ground.

The definition may sound like this: fog is the extension of cloud computing that consists of multiple *edge nodes* directly connected to physical devices.



Such nodes are physically much closer to devices if compared to centralized data centers, which is why they are able to provide instant connections. The considerable processing power of edge nodes allows them to perform the computation of a great amount of data on their own, without sending it to distant servers.

Fog computing is a mediator between hardware and remote servers. It regulates which information should be sent to the server and which can be processed locally. In this way, fog is an intelligent gateway that offloads clouds enabling more efficient data storage, processing and analysis.

One should note that fog networking is not a separate architecture and it doesn't replace cloud computing but rather complements it, getting as close to the source of information as possible.

The new technology is likely to have the greatest impact on the development of IoT, embedded AI and 5G solutions, as they, like never before, demand agility and seamless connections.

Existing system

The fogging approach has many benefits for the Internet of Things, Big Data and real-time analytics. Here are the main advantages of fog computing over cloud computing:

- **Low latency** (fog is geographically closer to users and is able to provide instant responses)
- **No problems with bandwidth** (pieces of information are aggregated at different points instead of sending them together to one center via one channel)
- **Loss of connection is impossible** (due to multiple interconnected channels)
- **High security** (because data is processed by a huge number of nodes in a complex distributed system)
- **Improved user experience** (instant responses and no downtimes satisfy users)
- **Power-efficiency** (edge nodes run power-efficient protocols such as Bluetooth, Zigbee or Z-Wave)

Cons of Fog Computing

The technology doesn't have any apparent disadvantages, but some shortcomings can be named:

- **A more complicated system** (fog is an additional layer in the data processing and storage system)
- **Additional expenses** (companies should buy edge devices: routers, hubs, gateways)
- **Limited scalability** (fog is not as scalable as cloud)

Fog Computing vs. Cloud Computing: Key Differences

Cloud vs. fog concepts are very similar to each other. But still, there is a difference between cloud and fog computing on some parameters. Here is a point-by-point comparison of fog computing and cloud computing:

1. Cloud architecture is centralized and consists of large data centers that can be located around the globe, a thousand miles away from client devices. Fog architecture is distributed and consists of millions of small nodes located as close to client devices as possible.
2. Fog acts as a mediator between data centers and hardware, and hence it is closer to end-users. If there is no fog layer, the cloud communicates with devices directly, which is time-consuming.
3. In cloud computing, data processing takes place in remote data centers. Fog processing and storage are done on the edge of the network close to the source of information, which is crucial for real-time control.
4. Cloud is more powerful than fog regarding computing capabilities and storage capacity.
5. The cloud consists of a few large server nodes. Fog includes millions of small nodes.

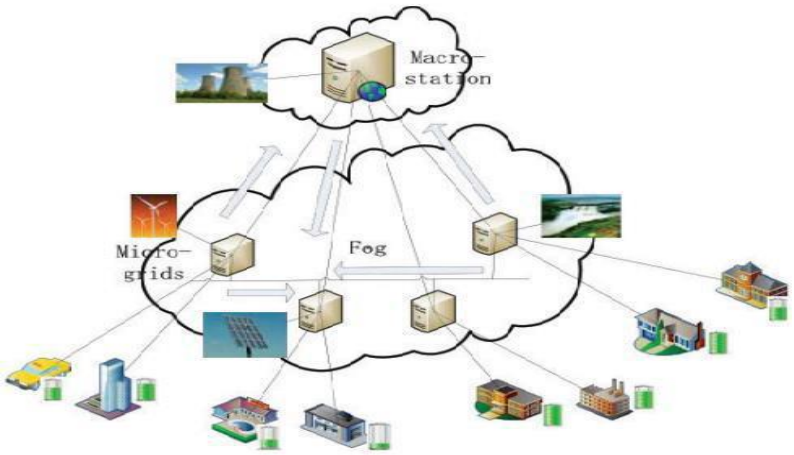
Cloud and Fog Computing: a Comparison Chart

	Cloud	Fog
Architecture	Centralized	Distributed
Communication with devices	From a distance	Directly from the edge
Data processing	Far from the source of information	Close to the source of information
Computing capabilities	Higher	Lower
Number of nodes	Few	Very large
Analysis	Long-term	Short-term
Latency	High	Low
Connectivity	Internet	Various protocols and standards
Security	Lower	Higher

(3) A brief review of the work already done in the field

In Smart Grid, fog computing is incorporated in order to automatically control the energy consumption. This is also three-tier architecture. Smart meter controls the Home Management Controller (HMC) and keep check and balance on the strict implementation of appliance scheduling. They both perform on first tier of fog computing. The second tier consists of utility control management that has the connection with smart meter in order to get information and meter recording of energy consumed. Consumers can also send their request and energy

demand to utility in order to get the services. The upper tier consists of cloud data centers that perform all the computation and storage related activities



Security issues can exist in different areas of FC. The most important areas where security issues can exist are networks, service infrastructure, and virtualization and user devices. Following table gives an overview of infrastructural analysis.

Table 1: Infrastructural Analysis of FC

Infrastructure of FC

Network	Service Infrastructure Core DCs Local DCs		Virtualization	User Devices
Man in the middle		Physical damage	Misuse of resources	Injecting information
Rogue Gateway		Privileges escalation		
DoS	Privacy leakage			
	Rogue DCs	VM manipulation		Service Manipulation
	Service Manipulation	DoS		

The next section presents the important aspects of security that can efficiently avoid and tackle the above mentioned security threats.

an offensive technique is presented to deny the data theft in FC environment. The author presented an intensive and lengthy procedure for

Authenticating the user for using CC and uploading/downloading the data. A series of questions followed by the login password are imposed for authenticating the user. Author proposed Decoy technology to keep the data secure in the case of data theft attack. Other attacks are not considered in this paper, which could hinder the data security.

(4) Noteworthy contributions in the field of proposed work

Policy driven security management system is proposed. After a critical analysis of FC architecture, the author presented a policy based management system for FC in order to make it secure and efficient. The proposed system consists of policy decision engine, application administrator, policy resolver, repository and enforcer. It is actually a combination of human defined polices and computer based system to enforce security. The proposed system can provide the mechanism of authentication, authorization, data security and integrity. The case study of smart transportation system is used and the simulated results show that the proposed system can provide a comprehensive security to the FC. The author neglected some other important aspects of security such as trust, fault tolerance and virtualization.

An interesting theoretical security technique is presented. This technique focuses the data security, user authentication and privacy of data and end users. The proposed technique comprises of decoy method along with some new modifications. The implementation strategy consists of fake nodes and fake data that consist of hidden batch files along with every legal node. All the illegitimate users are directed toward the fake node, these users download the fake data and when they run this data, the hidden batch files collect the identity information like MAC ID of the user. This collected information uniquely identifies the adversary and can help to take the action.

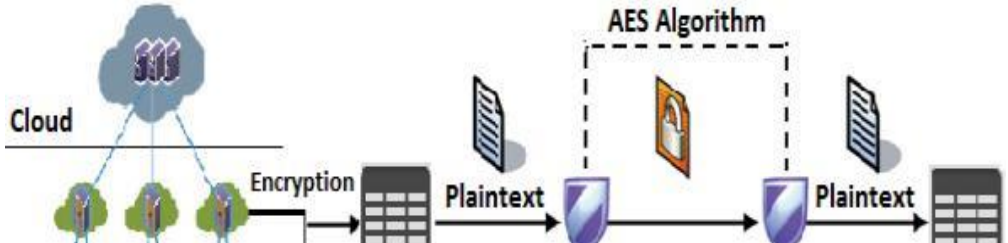
briefly discussed the security issues of FC. After discussing different implementation scenarios of FC, the author analyzed the case study of man in the middle attack. The memory consumption and CPU consumption of affected system is measured in the paper in order to show the effects of this particular attack on the speed and memory of the system. Solutions to prevent and recover from the attack are not covered in this paper.

(6) Proposed System

Data Encryption standard (DES) was once most widely used encryption standard, which uses symmetric key algorithm for encryption of data. This was considered to be basic building block for the advancement in the modern cryptography in present world. DES [12] has 56 bits of key size and whereas the block size is 64 bit. For many applications when considered DES is said to be the most insecure technique for many applications. This is because of its key size which is 56 bits and this could be brute forced. Two companies together had break the DES algorithm key in 22 hours and 12 minutes. This shows how weak the algorithm is. Some of the attacks that could break the key faster than the Brute force are Differential Cryptanalysis, Linear Cryptanalysis and Improved Davies Attack.

The predecessor of the DES algorithm is 3DES which is named as Triple Data Encryption Standard. Where 3 instances of DES are cascaded. The initial 56 bit key was sufficient, but the increase in computational power made brute force easy. Triple DES has made no changes to the previous DES algorithm except the increase in the key size, where it can have 56 or 112 or 168 bits of key size and whereas the block size remains same as 64 bits as DES. Triple DES was said to be 2½ time more secured than the DES algorithm. Even in Triple DES is vulnerable to security attacks meet in the middle attack. As DES algorithm was designed for hardware implementation, it is not reliable in hardware in the same way Triple DES do not function properly in software applications.

To overcome the above problem mentioned Advanced Encryption Standard (AES) is considered as more effective. Which is considered to be the most advanced and secured standard for encryption of electronic data. AES is considered to be successor of the DES which uses standard symmetric key encryption for many of the US federal organizations. AES accepts of the key size of 128, 192, 256 bits of size. Whereas 128 is already considered to be unbreakable and there were many open competition held by many organization to break the key but it was never done. On comparing all the available encryption algorithms, AES would be the better and most secured type of algorithm that could be implemented in the fog. So far encryption technique has not been proposed for security in the fog computing. As a conclusion over all the different type of encryption techniques, AES can be considered more suitable and adaptable for the environment of fog. Hence this paper includes applying of AES algorithm for security of the data in fog computing through an edge device of mobile.



1) **The SubBytes step:** The substitution byte of each byte could be found in lookup table. The size of lookup table is 16×16. Substitute byte for given input could be found by dividing the byte into two 4-bit pattern, resulting an integer value from 0 to 15. These could be represented by Hexadecimal values from 0 to F. Where one of it is used to find the row index and another is used for column index to get into the 16×16 Lookup table. In fig 2 each of the SubBytes step of dataset is replaced with the 8-bit lookup table. The Substitution step concentrates on reducing the correlation between input and output bits at byte level.

Algorithm 1:

```
Void SubByte(byte[][] state) {
        for (int rw=0; rw<4: rw++)
        for (int cl=0; cl<N; cl++)

state[rw][cl]=SBox[state[rw][cl]]; }
```

- *Step 1:* As in algorithm 1, initially the dataset are stored in the block.
- *Step 2:* Next the each of the block will be considered which has size of 256 bit.
- *Step 3:* Now each block is divided into two and considered as row and column value of S box.
- *Step 4:* Now the value is taken from the S box and the data is replaced by hexadecimal value.
- *Step 5:* Now the 1-4 steps are continued for all of the blocks in the same way.

2) **The ShiftRows step:** The most important matrix representation of the state array happens here as in Fig.3.The ShiftRow transformation behaves like. 1) It won't shift the state array at all in the first row.
2) Circularly second row will be shifted by one byte to the left. 3) In the third row circularly
shifting two bytes to the left. 4) In the fourth row it will circularly shift three bytes to left. In Shift Row step each of row will be swapped to its left depending on the index of row. In the same way for decryption, the corresponding rows will be shifted to opposite direction. The first row remains unchanged, in the second row the row will be shifted to right by one byte. Third row will be shifted to right by 2 bytes and in fourth row they are sifted to 3 bytes to right.

Algorithm 2:

```
Void ShiftRow(byte[ ][ ] state) { byte[ ] s= new byte[4]; for (int t=1;
t<4; t++)

        for (int d=0; d<N; d++) s[d]=state[t][(d+t)%N];

for (int d=0; d<N; d++)

        state[t][d]=s[d];

}

}
```

- *Step 1:* In algorithm 2, the hexadecimal values will be shifted to left, the row 1 will not be shifted.
- *Step 2:* In the row 2 it will be shifted to 1 byte to left, the loop will be continued until all of the blocks in the row are shifted to left.
- *Step 3:* In row 3 the block will be shifted to 2 byte left and continued for all of the bytes in the row.
- *Step 4:* In row 4 the block will be transferred to left by 3 bytes and the same process is continued.

3) **The MixColumns step**: In Mix Column each byte of the column in dataset is replaced with function of all bytes in the existing column as in Fig 4. And more importantly, each byte in the column will be replaced by the two times of that byte, plus three times of next byte, plus the byte that comes next, plus the byte the follows.

(6) Results: Expected outcome of the proposed work

several security and privacy issues in the context of fog computing, which is a new computing paradigm to provide elastic resources at the edge of network to nearby end users and also in data storage management system.

- we discuss security issues such as secure data storage, secure computation and network security.

- We also highlight privacy issues in data privacy, usage privacy, and location privacy, which may need new think to adapt new challenges and changes

References

[1] A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and

Cryptography against memory attacks. In *Theory of Cryptography Conference (TCC)*, pages 474–495, 2009.

[2]. C. Basescu, C. Cachin, I. Eyal, R. Haas, and M. Vukolic. Robust Data Sharing with

Keyvalue Stores. In *Proceedings of the 30th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, PODC '11, pages 221–222, New York, NY, USA, 2011. ACM.

[3]. A. Beimel. Secret-sharing schemes: A survey. In *Third International Workshop on Coding and Cryptology (IWCC)*, pages 11–46, 2011.

[4]. A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa. DepSky: Dependable and

Secure Storage in a Cloud-of-clouds. In *Proceedings of the Sixth Conference on Computer*

Systems, EuroSys '11, pages 31–46, New York, NY, USA, 2011. ACM.

[5]. V. Boyko. On the Security Properties of OAEP as an All-or-nothing Transform. In

Procedings of CRYPTO, pages 503–518, 1999.

[6]. R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky. Deniable Encryption. In *Proceedings of*

CRYPTO, 1997.

[7]. Cavalry. Encryption Engine Dongle. http://www.cavalrystorage.com/

en2010.aspx/.

[8]. C. Charnes, J. Pieprzyk, and R. Safavi-Naini. Conditionally secure secret sharing schemes with disenrollment capability. In *ACM Conference on Computer and Communications Security (CCS)*, pages 89–95, 1994.

[9]. A. Desai. The security of all-or-nothing encryption: Protecting against exhaustive key search. In *Advances in Cryptology (CRYPTO)*, pages 359–375, 2000.

[10]. Y. Dodis, Y. T. Kalai, and S. Lovett. On cryptography with auxiliary input. In *ACM Symposium on Theory of Computing (STOC)*, pages 621–630, 2009.

[11]. C. Dubnicki, L. Gryz, L. Heldt, M. Kaczmarczyk,W. Kilian, P. Strzelczak, J. Szczepkowski, C. Ungureanu, and M. Welnicki. HYDRAstor: a Scalable Secondary Storage.

In *FAST'09: Proceedings of the 7th USENIX Conference on File and Storage Technologies*, pages 197–210, Berkeley, CA, USA, 2009. USENIX Association.

[12]. M. Duermuth and D. M. Freeman. Deniable Encryption with Negligible Detection Probability: An Interactive Construction. To Appear in EUROCRYPT, 2011. Available from

http://eprint.iacr.org/2011/066.pdf.

[13]. EMC. Transform to a Hybrid Cloud. http://www.emc.com/campaign/global/

hybridcloud/index.htm.

[14]. IBM. IBM Hybrid Cloud Solution. http://www-01.ibm.com/software/

tivoli/products/hybrid-cloud/.

[15]. ITIF. How Much Will PRISM Cost the U.S. Cloud Computing Industry? http://www2.

itif.org/2013-cloud-computing-costs.pdf.

[16]. M. Klonowski, P. Kubiak, and M. Kutylowski. Practical Deniable Encryption. In *Proceedings of SOFSEM: Theory and Practice of Computer Science*, 2008.

[17]. H. Krawczyk. Secret Sharing Made Short. In *International Conference on Advances in*

Cryptology, 1993.

[18]. L. Lamport. On interprocess communication, 1985.

[19]. S. Micali and L. Reyzin. Physically observable cryptography (extended abstract). In *Theory of Cryptography Conference (TCC)*, 2004.

[20] NEC Corporation. HYDRAstor Grid Storage System. http://www.hydrastor.com.