A Unique Approach to Element Management and Secure Cloud Storage Backup for Sensitive Data

Srinivas M N
Celstream Technologies Private Limited, Bangalore

Srinivas B V
Dept of CSE, EPCET, Bangalore

Marx R
Software group, B.E.L, Bangalore

ABSTRACT
With dynamic trends and new developments in hardware and software, and the need to manage them efficiently, designing an Element Management System (EMS) for effective management of Network Element's (NE) is a challenging task. Small and medium businesses may not have the required skills or resources to manage their EMS. Large businesses may also want to use EMS services from skilled vendors. With the emergence of cloud computing technology, EMS can be provided as a service, i.e., SaaS (software as service) to different customers. As promising as it is, this paradigm also brings forth many new challenges for data security and access control when users outsource confidential and old management data for sharing on public cloud ,i.e., which are not within the same trusted domain. In this proposed scheme, the EMS is provided as a service to different sites of the organization using a private cloud. Sensitive data is stored in the public cloud which comprises of the current configuration of the complete private cloud, important disaster recovery data and management data. This would provide a confidential and secure backup of EMS. This would be useful in the case of natural calamities (tsunami, tornado, earth quake, etc.), terrorist attacks and any other disaster that strikes the private cloud. The clients/data owners, who require management data, can access the data from the public cloud. While accessing data from the public cloud, integrity and confidentiality of the data is preserved by using 2 techniques, namely Attribute Based Encryption (ABE) and Proxy Re-Encryption (PRE).

Keywords
Element Management System, Network Element, Disaster recovery, Attribute Based Encryption, Proxy Re-Encryption, Disaster Recovery plan, mission critical functions.

1. INTRODUCTION
EMS is concerned with managing network elements of the same type. In the current trends and ever changing technologies, providing efficiency and meeting the demands of availability, reliability and scalability is a difficult task and it requires a lot of effort and dedication from the developer [1]. At present with the emergence of cloud computing paradigm resources of the computing infrastructure are provided as services over the Internet, i.e., Software as a service (Saas), Hardware as a service (Haas) and Platform as a service (Paas). Successful examples are Amazon's EC2 and S3 [12], Google app Engine [13] and Microsoft Azure [14] which provide users with scalable resources at relatively low prices. For example, Amazon's S3 data storage service just charges $0.12 to $0.15 per gigabyte month.

In the past EMS for campus were developed in object oriented languages such as C/C++ which gives faster speed of execution but lack portability and state of the art features and memory leaks are unavoidable and it uses operating system-dependant threads. Less focus was given on alarm generation and alarm diagnosing, interfaces developed earlier were less attractive and statistics and reports generated were not sufficient for future analysis. Element management has a high priority in small enterprises but the development is limited due to lack of investments and dedicated personnel [1]. [2] To ensure confidentiality of sensitive data against untrusted servers, existing solutions usually apply cryptographic methods by disclosing data decryption keys only to authorized users. In doing so, these solutions inevitably introduce a heavy computation overhead on the data owner for key distribution and data management. Disaster recovery has been ignored by many organizations citing prohibitive costs. Disaster recovery is an important aspect of enterprise computing. A disaster recovery plan (DRP) describes how an organization is to deal with potential disasters, i.e., it consists of the precautions taken so that the effects of a disaster will be minimized and the organization will be able to either maintain or quickly resume mission critical functions.

In the proposed scheme, The EMS is provided as a service to different vendors using a private cloud and public cloud is used as a disaster recovery site citing costs. [1] The EMS is developed using The Java platform, as it's simple, easy to code, portable, avoids memory leaks, multithreaded, has a good feature set and a rich set of API's and is an open source technology. The management protocol used is SNMPv2C. This version is chosen as it provides improved error handling and improved set of commands than the previous versions. SNMP4J API is used as it is an open source API, it has a good set of functions and ample examples are available on web.

Disaster recovery is an important and essential aspect of any enterprise. According to the Disaster Recovery Journal, an estimated $78,000 per hour is lost due to system downtime. So having calculated the losses incurred due to various natural disasters in the absence of such a service, approximately 40% of all businesses have now made disaster recovery services an essential part of their systems. As devices, systems, and networks become ever more complex, there are simply more things that can go wrong. As a consequence, recovery plans have also become more complex. Most businesses today are heavily automated and damage to equipment or data can not only disrupt business continuity and inflict financial losses but also threaten the very survival of a company. There is a need finding new ways to keep data up and running and still beat the expenses.

In the proposed scheme we store the sensitive data, i.e., (access permissions, IP addresses ,port numbers of all the elements in the network, statistics and graphs on performance data, packet loss, elements system description, traffic data, alarms and traps) and the complete configuration of the private cloud in the public cloud which acts as a disaster recovery site. This disaster recovery site provides a host of

services to ensure business continuity through restoration of data. A geographically distributed public cloud infrastructure is better as it would naturally provide more protection to the data and configuration. [2] For the purpose of helping the data owners enjoy fine-grained access control of data stored on untrusted cloud servers and to reduce the heavy computation overhead on him/her our proposed scheme enables the data owner to delegate tasks such as data file re-encryption and user secret key update to public cloud servers without disclosing data contents or user access privilege information. We achieve this goal by exploiting a novel cryptographic primitive, namely Key Policy Attribute-Based Encryption (KP-ABE) [15] and uniquely combine it with the technique of PRE [16]. In this scheme the user need not online at all times.

This article is structured as follows:

The background and concise description of the EMS, EMS in cloud, Fine grained data access control and importance of disaster recovery is presented in section 1. Section 2 discusses models and assumptions. Section 3 reviews some technique preliminaries related to our work. Related work is presented in section 4. The Structural design and technical framework of the EMS is presented in section 5 and 6. Our proposed scheme is presented in section 7. The work is concluded in section 8. Enhancement to the work is also proposed in section 8.

2. MODELS AND ASSUMPTIONS

2.1 System Models

We assume that the system is composed of the following parties: the Data Owner (Administrator of the private cloud), many clients within and outside the scope of the private cloud who request EMS as a service, many trusted and untrusted Cloud Servers, and a Third Party Auditor if necessary. Only authorized persons with admin privileges can request EMS as a service. The management data is stored on the trusted cloud and can be accessed by clients. Old management data and the current configuration of the complete private cloud and important disaster recovery data are stored in the public cloud citing costs. Public cloud infrastructure is better as it would naturally provide more protection to the data and configuration. This would be useful in case any disaster that strikes the private cloud. Clients can access the old management data and download data files of their interest from public cloud servers directly without the intervention of the private cloud and then decrypt. Neither the data owner nor users will be always online. They come online just on the necessity basis. Both the trusted and untrusted cloud Servers are always online. The private cloud had limited storage capacity and computational power whereas the public cloud is assumed to have abundant storage capacity and computation power. The Third Party Auditor is also an online party which is used for auditing every file access event.

2.2 Security Models

In this work, trusted cloud servers will follow our proposed protocol strictly whereas untrusted cloud Servers will follow our proposed protocol in general, but try to find out as much secret information as possible based on their inputs. More specifically, we assume untrusted cloud servers are more interested in management data/file's contents and user access privilege information than other secret information. [2] Untrusted cloud servers might collude with a small number of malicious users for the purpose of harvesting management data/file's contents when it is highly beneficial. Communication channel between the data owner/users, trusted and untrusted cloud servers are assumed to be secured under existing security protocols such as SSL. Users would try to access management data/files either within or outside the scope of their access privileges. To achieve this goal, unauthorized users may work independently or cooperatively. In addition, each party is preloaded with a public/private key pair and the public key can be easily obtained by other parties when necessary.

3. TECHNICAL PRELIMINARIES

3.1 Key Policy Attribute-Based Encryption (KP-ABE)

KP-ABE [15] is a public key cryptography primitive for one-to-many communications. In KP-ABE, data are associated with attributes for each of which a public key component is defined. The encryptor associates the set of attributes to the message by encrypting it with the corresponding public key components. Each user is assigned an access structure which is usually defined as an access tree over data attributes, i.e., interior nodes of the access tree are threshold gates and leaf nodes are associated with attributes. User secret key is defined to reflect the access structure so that the user is able to decrypt a ciphertext if and only if the data attributes satisfy his access structure. Please refer to [15] for more details on KP-ABE algorithms.

3.2 Proxy Re- Encryption (PRE)

Proxy Re-Encryption (PRE) is a cryptographic primitive in which a semi-trusted proxy is able to convert a ciphertext encrypted under A's public key into another ciphertext that can be opened by B's private key without seeing the underlying plaintext. More formally, a PRE scheme allows the proxy, given the proxy re-encryption key $rk_{a<->b}$, to translate ciphertexts under public key pk_a into ciphertexts under public key pk_b and vice versa. Please refer to [16] for more details on proxy re-encryption schemes.

4. RELATED WORK

Existing work close to ours can be found in the areas of 'Network management system framework', 'Element management' and 'Secure, scalable and fine grained data access control in cloud computing'.

[3] Xiaosong Wang proposed a basic network management system framework of campus network based on Web and its deployment method. This covers most of the demand for network management. This framework has a good scalability, which lays the foundation for the development of Network Management System in campus network.

Srinivas.M.N proposed an EMS to manage NEs in the campus like environment. The EMS addresses primary issues of element management. The implementation is web based, provides ease of access and is robust in nature. The implementation uses open source technologies, open source API's and lower cost tools. In case of client machines (hosts), the SNMP service facilitated by the OS is used, resulting in enhanced efficiency.

In this implementation, when a new element is added, a separate instance of the functional module is created for that element. This improves scalability as any number of new elements can be added easily. The solution is implemented using Java and is highly portable, i.e., the solution can be used without modification on multiple platforms. Data integrity is

preserved by the solution implementation. This implementation exposes its functionalities through a well-defined API and can be extended to be compatible with systems from different vendors. Interactive User Interface (UI) and customized event viewing provided by this implementation enable the user to use the deployed system effectively.

Hwa-Chun Lin proposed a Web-based distributed network management architecture in which the HTTP protocol is used for a network management console to access network management applications and for network management applications located at different sites to communicate with each other. No other protocol is used for communications between the console and network management applications or between the network management applications.

The major advantage of using the same protocol (HTTP) for console-application and application-application communications is that the network management server requires minimum system resources such as computation power, memory, disk space, and etc. This is particularly important when a network management server is to be embedded in a device.

[4] Md. Jakir Hossen proposed the architecture for Web based Network Configuration Management, which is capable of controlling and monitoring the configuration information of enterprise networks remotely. This paper also discusses the functional requirements of Web-based Network Management and also explains the function and design of prototype network device management station based leading-edge Java servlet technologies and the World Wide Web.

[2] Shucheng Yu proposed a scheme for Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing. The problem of simultaneously achieving fine-grainedness, scalability, and data confidentiality of access control is addressed by defining and enforcing access policies based on data attributes. The burden on the data owner is reduced by delegating most of the tasks to untrusted cloud servers. This is achieved by exploiting and uniquely combining techniques of attribute-based encryption (ABE), proxy re-encryption, and lazy re-encryption.

Ateniese et al [5] proposed a secure distributed storage scheme based on proxy re-encryption. Specifically, the data owner encrypts blocks of content with symmetric content keys. The content keys are all encrypted with a master public key, which can only be decrypted by the master private key kept by the data owner. The data owner uses his master private key and user's public key to generate proxy re-encryption keys, with which the semi-trusted server can then convert the ciphertext into that for a specific granted user and fulfill the task of access control enforcement. In [7], Vimercati et al proposed a solution for securing data storage on untrusted servers based on key derivation methods.

5. OVERALL STRUCTURAL DESIGN OF EMS

The overall structure as in Figure 1 depicts n-tier client/server architecture, where the EMS is depicted as server and managed objects as clients. In the proposed structure, any number of managed objects can be handled by the server. As exposed in Figure 1, the EMS structure is split into presentation layer, management layer and data layer.

5.1 Presentation layer

The main function of this layer is to obtain user inputs, create tasks for the system, and translate tasks and results to something that a user can understand.

5.2 Management layer

The core functionality of the EMS exists in the form of Plain Old Java Classes in the management layer. The EMS is based on a centralized structure which facilitates improved security, consistency and accuracy of management data. The management functions which are required for the efficient functioning of the EMS are complemented by the ISO defined network management tasks as shown in Figure 1.The sub modules in the management layer are: fault management, system management, performance management, traffic Management, configuration management and security management [3].

5.3 Data Layer

Data layer includes functions to effectively store and retrieve data from the database server in cloud. Data integrity is also maintained. The administrator is the sole person to modify the data. Access to database server by unauthorized users is restricted. The data stored in the database server includes, access permissions IP addresses and port numbers of all the elements in the network. Statistics and Graphs on performance data, packet loss, element's system description, traffic data, alarms and traps received are also accumulated in the data layer.

6. EMS TECHNICAL ARCHITECTURE

The EMS is provided as a service by the Management server in cloud. The clients can avail the EMS by subscribing to the cloud. As in Figure 2, the web interface is built using JSP, XML and web service technology .The data collected may be network data, performance data, fault data , logs graphs, etc.,. The management layer provides the core functionality of the EMS. Customized event viewing and help document prove valuable to a new user.

The confidential data is stored in the database server. The interface used between the core functionality and the database server is JDBC, which is an open source technology. The web application was made attractive by using buttons and images developed using Flash CS4 .The image galleries were developed using CSS. The embedding of flash developed buttons and images in the application increased look and feel of the application to a great extent. SNMP4J is the API used by the EMS to communicate with the managed object [1].

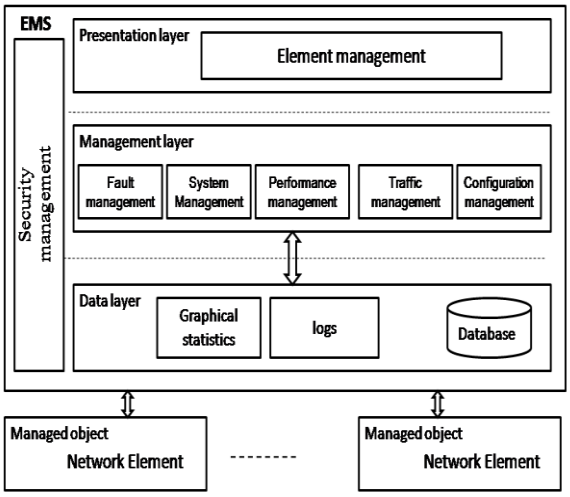

Fig 1: Structure of Element Management System

7. OUR PROPOSED SCHEME

7.1 Main Idea

Currently many applications are provided as service by the cloud. Our main idea is to provide EMS as a service. The EMS provides an efficient structural design and technical architecture. The essential features provided by EMS are as follows.

Multi level thresholding and advanced alarm diagnosing, which describes alarms in full detail, severity of the alarms with suggestions for probable causes and remedies. Real time management data is reflected in the form of statistics and graphs which can be helpful in future analysis and research. They are available to the user even when the application is not running. A vital role is played by the statistics and data collected in future enhancements. EMS is deployed redundantly and can scale to support hundreds of concurrent users by simply adding processing power [1]. A help document that provides overview of EMS and brief instructions on installation of the EMS has been created. Functionality of the EMS is explained visually with the help of graphical data [1].

As in Figure 3, clients within the organization and outside the organization can make use of the EMS from the private cloud.

The present management data is available to the clients directly from the private cloud. In order to save resources the private cloud stores the old management data in the public cloud. The public cloud also acts a disaster recovery site. The public cloud infrastructure is in a different geographical location which provides more protection to data and configuration. This would be helpful if any disaster strikes the cloud. The management data outsourced is very confidential and is encrypted by the Administrator of the private cloud. In case clients want to access the old management data from the public cloud fine grained access control is required.

In order to achieve secure, scalable and fine-grained access control on outsourced data in the cloud, we utilize and uniquely combine KP-ABE and PRE[2]. We associate each data file with a set of attributes, and assign each user an expressive access structure which is defined over these attributes. This technique alone incurs a lot of burden on the data owner, i.e., who uploads the data. To address this issue we uniquely combine PRE with KP-ABE and enable the data owner to delegate most of the computation intensive operations such as re-encryption of all the data files accessible to the leaving user and updating of secret keys for users of cloud servers without disclosing the underlying file contents. Such a construction allows the data owner to enforce access control on data files with a minimal overhead in terms of computation effort and online time.

Data confidentiality is maintained as the data cannot be accessed in plaintext by the untrusted servers. A third party auditor is used to maintain the list of access events

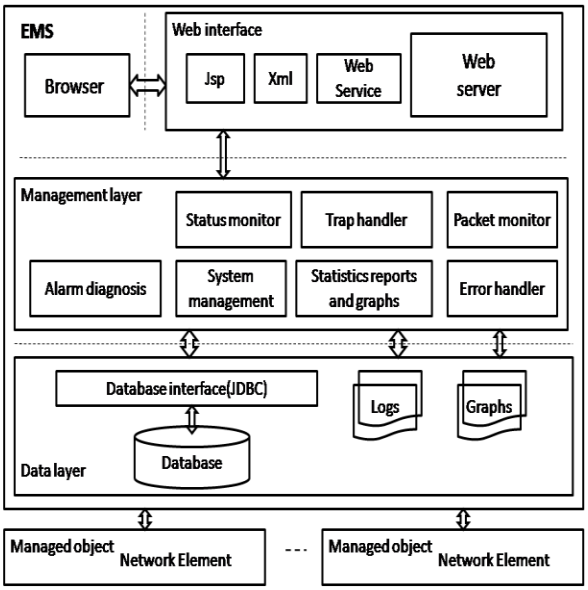

Fig 2: Technical architecture of EMS

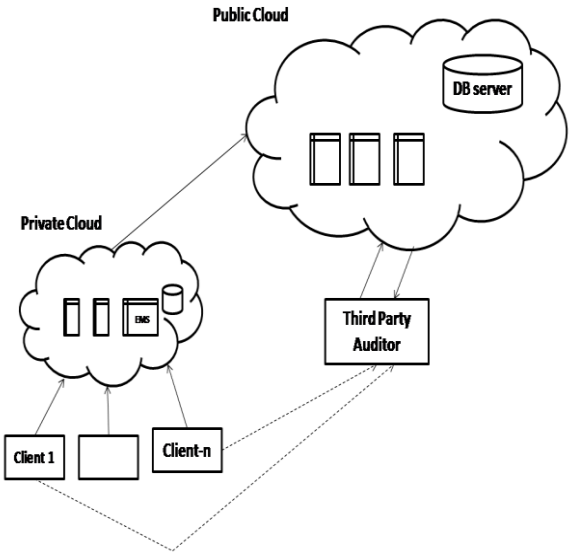

Fig 3: Proposed architecture

8. CONCLUSION AND FUTURE SCOPE

The primary focus of this work is to provide mission critical applications such as EMS as a service. The importance of disaster recovery and how small enterprises can use the existing public infrastructure as disaster recovery site is also presented in this work. Fine grained access control and data integrity and security are achieved by uniquely combining KP-ABE and PRE. In future the author is planning to provide other mission critical applications as a service. Further the author is trying to develop cloud management tools to ensure that, the cloud computing-based resources are working optimally and interacting well with users and other services.

9. REFERENCES

[1] Srinivas.M.N," An Efficient and Cost-Effective Approach to Manage Network Elements in a Campus like Environment", International Journal of Advances in Computer Networks and Security, pp.28-32, 2011.

[2] Shucheng Yu,"Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing".

[3] Xiaosong Wang, "Studies on Network Management System Framework of Campus Network", 2nd International Asia Conference on Informatics in Control, Automation and Robotics, pp. 285-289, Car 2010.

[4] Md. Jakir Hossen, Abd Rahman Ramli, and Mohd. Khazani Abdullah,"Web-based Network Device Management Using SNMP Servlet",IEEE2003.

[5] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," in Proc. of NDSS'05, 2005.

[6] Hwa-Chun Lin and Chien-Hsing Wang,"Distributed Network Management by HTTP-based remote invocation", Global Telecomrnunications Conference – Globecorn '99.

[7] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-encryption: Management of access control evolution on outsourced data," in Proc. of VLDB'07, 2007.

[8] Chien-Chung Shen,"A Network Management Architecture for Battlefield Networks" ATIRP, pp. 1226-1231, IEEE 1997.

[9] G. Mansfield, M. Murtha, K. Higuchi, K. Jayanthi, B.Chakraborty, Y. Nemoto and S.Noguchi," Network Management In a Large-scale OSI-based Campus Network using SNMP", IEEE 1992.

[10] Hwa-Chun Lin and Chien-Hsing Wang,"Distributed Network Management by HTTP-based remote invocation", Global Telecomrnunications Conference – Globecorn '99.

[11] Jae-Kyu Chun, Ki-Yong Cho, Seok-Hyung Cho, Young-Woo Lee and Young-Il Kim," Network Management Based on PC Communication Platform with SNMP and mobile agents" Proceeding of the 22nd international on Distributed systems workshop,(1CDCSW'02),IEEE 2002.

[12] Amazon Web Services (AWS), Online at http://aws. amazon.com.

[13] Google App Engine, Online at http://code.google.com/appengine/.

[14] Microsoft Azure, http://www.microsoft.com/azure/.

[15] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. Of CCS'06, 2006.

[16] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in Proc. of EUROCRYPT '98, 1998.

[17] Annie Ibrahim rana,"New Roles of Policy –based Management in Home Area Networks-Concepts, Constraints and Challenges",IEEE 2009.

[18] Jong-Wook Beak," ATM Customer Network Management Using WWW and CORBA Technologies". IEEE 1998.

[19] Marcus Burner,"Probabilistic Decentralized Network Management " , IEEE 2009.

[20] Harry Li and Guangjing Chen," Wireless LAN Network Management System",IEEE 2004.

[21] WebNMS Online at http://www.webnms.com

[22] OpenNMS Online at http://www.opennms.com

[23] SNMP4J API Online at http://www.snmp4j.org/

[24] Cloud Online at http://thecloudtutorial.com/