Ex-HABE with User Accountability for Secure Access Control in Cloud

Nilesh S. Jadhav, Ms. D. A. Chaudhari, DYPCOE - Akurdi, Savitribai Phule Pune University

Abstract—Data outsourcing is becoming a useful and feasible paradigm with the rapid application of service-oriented technologies. Many researchers have tried combination of access control and cryptography to propose a model to protect sensitive information in this outsourcing scenario. However, these combinations in existing approaches have difficulty in key management and key distribution when fine-grained data access is required. Taking the complexity of fine-grained access control policy and the wide-reaching users of cloud in account, this issue would become extremely difficult to iron out. Various system models using attribute-based encryption (ABE) have been proposed however, most of them suffer from heavy overhead in implementing the access control policies. In this paper, a system is proposed with extended hierarchical attribute-based encryption (HABE) by using ciphertext-policy attribute-based encryption (ABE). It uses the hierarchical structure of users and bilinear mapping for generating the keys for various data handlers. Also the system focuses on user tracking by allocating an unique id to user. The system uses traitor tracing along with separation of duty made available by HABE and reduces the scope of key abuse. It is formally proved extended HABE with traitor tracing adds on to user accountability if user tracking for resource is maintained for hierarchical systems.

Index Terms—Cloud Computing; access control; attribute- based encryption; hierarchical; bilinear mapping; user tracking; traitor tracing

*****

I. INTRODUCTION

Cloud computing is a significant preferment in the delivery of information technology and services. Today, cloud computing is the progression and convergence of several trends that have been driving enterprise data centers and service providers over the last several years [1]. One definition that is frequently drawn upon by experts is that of the USA's National Institute of Standards and Technology (NIST):
"Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources(e.g. networks, servers, storage, application and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [2]."
Although the great benefits brought by cloud computing paradigm are rousing for IT companies, academic researchers, and potential cloud users. At the same time security problems in cloud computing become serious obstacles which, without being appropriately deal with, will avert cloud computing's extensive applications and usage in the future. One of the noted security concerns is data security and secrecy in cloud
Computing due to its networked data storage and management. In cloud computing, users have to upload their data to the cloud service provider (CSP) for storage according to the service model used. The concern is with accordance to the cloud service provider as it is a commercial enterprise which cannot be totally trusted. This is the foremost data security requirement.
Along with data confidentiality flexible and fine-grained access control is also strongly desired in the service-oriented cloud computing model. Considering health-care information system [6], on a cloud is required to restrict access of secured medical records to worthy doctors. A customer relation management (CRM) system running on a cloud may allow access of customer information to high-level executives of the company only. In these systems, access control of sensitive data is either required by legislation (e.g., HIPAA) or company regulations.
Endorsement in cloud environment is needed in order to protect cloud resources and restrict unauthorized access to them. In some systems it is based on encryption mechanism in which access is restricted in such a way that the data owner (DO) reveals decryption key only to those users having the required attributes for the file being accessed. There are various approaches for access control in cloud computing as discussed in upcoming sections.

II. RELATED WORK

In this section, a review to the notion of attribute-based encryption (ABE) is made to provide a brief overview of Bobba et al. scheme. After that, a survey is made on existing secure access control schemes based on ABE.

A. Attribute-Based Encryption

The idea of ABE was first instigated by Sahai and Waters [1] as a new method for fuzzy identity-based encryption. The primary drawback of the scheme in [1] is that its threshold semantics lacks expressibility. Number of efforts are followed in the literature to try to solve this problem. In the ABE strategy, ciphertexts are not encrypted to one specific user as in traditional public key cryptography. Rather, both ciphertext and user decryption keys are associated with a set of attributes or a policy over attributes. A user is able to decrypt a ciphertext only if there is a equivalence between his decryption key and the ciphertext.
ABE schemes are categorized into key-policy attribute based encryption (KP-ABE) and ciphertext-policy attribute based ncryption (CP-ABE), be resultant from how attributes and policy are related with ciphertexts and users decryption keys. In a KP-ABE scheme [2], a ciphertext is kindred with a set of attributes and a user's decryption key is linked with tree access structure. Only if the attributes associated with the ciphertext satisfy the user tree access structure, he can decrypt the ciphertext. In a CP-ABE scheme [3], the parts of ciphertexts and decryption keys are interchanged; Encryptor selects the tree access policy according to which the ciphertext is encrypted, while the corresponding decryption key is created with respect to a set of attributes. The key can be used to decrypt the ciphertext till the set of attributes associated with a decryption key meet the tree access policy associated with a given ciphertext. Since user decryption key is associated with a set of attributes, CP-ABE is conceptually closer to traditional access control models such as Role-Based Access Control (RBAC) [3]. Considering this it is more natural to apply CP-ABE, instead of KP-ABE to impose access control of encrypted data.
In a CP-ABE scheme, the policy is defined by considering

various attributes and their combinations. User can use all combinations of attributes in single set armed in their keys to satisfy policies. Bobba et al. [4] solved this problem by introducing ciphertext-policy attribute-set-based encryption (CP-ASBE).CP-ASBE is commonly termed as ASBE which is widened form of CP-ABE which organizes user attributes into a recursive set structure. ASBE can impose dynamic constraints on combining attributes to satisfy a policy, which provides great flexibility in access control. In [4], several possible solutions with plain CP-ABE are described, but none of them is satisfactory. However, using ASBE, the issue can be handled simply by allocating multiple values to the group of attributes in different sets. Furthermore, ASBEs capability of assigning multiple values to the same attribute enables it to solve the user revocation problem efficiently, which is difficult in CP-ABE. The revocation problem can be solved easily by adopting the policy of expiration times.

B. Secure Access Control Solutions for Cloud Computing

The traditional method to fortify sensitive data outsourced to third parties is to pile encrypted data on servers and the decryption keys are disclosed to authorized users only. There are several drawbacks of this trivial solution. Various solutions proposed need efficient key management mechanism to manage decryption keys to authorized users, which has been proven to be very difficult. Also DO need to be online all the time so as to perform various tasks associated like encrypt, distribute keys to authorize users.

ABE is an agreeable solution for achieving secure, scalable, flexible and fine-grained access control solutions. Yu et al. [5] presented an access control mechanism considering KP-ABE. The scheme jointly used re-encryption technique for well-organized user revocation and allowed the data owner to transfer most of the computational work to cloud servers. Graceful fine-grained access control was achieved by using KP-ABE.

There are various problems with Yu et al. scheme. The encryptor is not able to settle who can decrypt the encrypte data except choosing descriptive attributes for the data. The encryptor has to trust the key issuer. Also, KP-ABE used in the scheme is not suitable to certain applications. For such a request, a better option is CP-ABE.

Wang et al. [16] proposed hierarchical attribute-based encryption (HABE). It combined hierarchical identity-based encryption (HIBE) and CP-ABE. This scheme supported both fine-grained access control and computation delegation to the cloud providers. However, HABE considered attribute administration by the domain master. Hence, the same attribute may be administrated by multiple domain masters due to various specified policies, which is difficult to implement in practice. Compared with ASBE, it doesn't support compound attributes effectively and multiple value conveyance.

Fig. 1: Proposed system Architecture

The CSP administer a cloud to provide data storage service. Data owners encrypt their data files and upload them in the cloud for sharing with data consumers. To access the shared data files, DCs download encrypted data files of their interest from the cloud and then decrypt them. Each DO/DC is managed by a domain authority. A domain authority is ad- ministered by its parent domain authority or the TA. DO, DC, DA and the trusted authority are organized in a hierarchical manner as shown in Fig. 1.

The TA is the root authority and responsible for managing top-level domain authorities. Each top-level domain authority corresponds to a top-level firm, such as a education institution, while each lower-level domain authority corresponds to a lower-level organization, such as an associated department in a institution. Data owners/consumers may be analogous to employees in an organization. Each domain authority has to supervise

C. ABE User Accountability approaches

Various researchers have tried to address the problem of achieving fine-grained access control with efficient user revocation and accountability. Any system trying to address the issue focuses on basic algorithmic steps KeyGeneration, Encryption and Decryption. An attempt made by using traitor tracing algorithm faced a serious efficiency drawback in scalable system. Also it faces the issue of linear complexity rise for linear rise of authorized users.

The issue was considered and monitored by using ABE which also follows the same algorithm steps. In ABE, number of attributes are considered then the number of users while deciding the length of public parameters and ciphertext as in most of the other techniques. Thus, ABE enables public key based one-to-many encryption, where differential yet flexible access rights can be assigned to individual users. Use of CP-ABE allows to specify access policy for the same file for various different attributes according to the DC need.

III. IMPLEMENTATION DETAILS

A. HABE

In this system specific IDs are given across each level of hierarchy. Also various attributes are given to the respective domains which adds on to the various next level addition and separation of sub-domain users. Most focusing aspect is of following separation of duty principle. However due to this principle it has concerns towards use in cloud environments but at the same time gives as flexibility privilege towards user revocation and accountability issue. This principle allocates the task for domain masters to handle the operations of attribute management and administration. As tasks and processes are divided across the system complexity to specify access control policies is higher. Use of bilinear mapping for key generation and tree access policy are the major enforcement mechanisms. These operation affect the system performance and efficiency. Also we have defined method to handle the policy conflicts in this proposed hierarchical system. Scope of the technique is limited to organizations having hierarchical structure of system entities. System tasks and processes are distributed at each node which is important to achieve separation of duty.

B. System Overview

As represented in Fig. 1, the cloud system considered consists of five types of parties: a Cloud Service Provider, Data Owners, data consumers (DC), a number of Domain Authorities (DA), and a Trusted Authority (TA). Following figure 1 depicted the overall architecture for proposed system:

domain authorities at the next level or the data owners/consumers in its domain.
In proposed system, neither data owners nor data consumers will be always online. They come online only when they have to upload or download the file respectively, while the CSP, TA, and DA are always online. The cloud is assumed to have huge storage capacity and computation power. In addition, it is assumed that data consumers can access data files for reading only.
Hierarchical structure defined as in Fig. 1, each body is associated with a public key and a private key, with the latter being kept secretly by the body. The TA acts as the root of trust and authorizes the top-level domain authorities. A DA is trusted by its subordinate domain authorities or users that it administrates, but may try to get the private keys of users outside its domain. Users may try to access data files either within or outside the horizon of their access privileges, so malicious users may conspire with each other to get sensitive files beyond their privileges. e proposed User Accountable ABE scheme seamlessly extends the HABE scheme to handle the hierarchical structure of system users. The TA has the responsibility of generating and distributing system parameters and root master keys as well as authorizing the top-level domain authorities. A domain authority is responsible for delegating keys to subordinate domain authorities at the next level or users in its domain. Each user in the system is allotted with a key structure which has an defined attribute associated with the users decryption key.

The toolkit provides a number of command line tools as follows:

1) *uaabe-setup:* Generates a public key Pk and a master key Mk (0). Setup algorithm is like an initialization algorithm where, public parameter and secret key is generated using bilinear mapping. Public parameters are broadcasted so that anyone can read them and secret key is kept private with authority to avoid Intruder attacks. Also the authority has the power for allocatoin and deallocaton of resource.
2) *uaabe-keygen:* Given Pk and Mk (0), generates a private key for a key structure. The private key is generated using Pk, Mk (0) and attribute for which the DO wants to upload or download the file.
3) *uaabe-File upload Scenario:* DO uploads a file F to the cloud servers, he first defines an access policy for this file. Each policy is defined according to the various attributes of the file. The policy defined is encrypted by using key generated by an symmetric key algorithm. After the needful the DO uploads the file.
4) *uaabe-DC Access:* Every DC willing to access the file is first authenticated under the TA. After authentication it is allocated a unique id. According the specifications he also has an access policy under the hierarchical TA/DA. According to the policy specified TA and DA validator is allocated to the DC. If the policy is satisfied for the specific user then only it is provided with the key access to read the secured file. The trusted authority keeps track of the file access across various levels to identify key abuse section across the system.
5) *uaabe-User accountability:* A user is provided with a hierarchical validator according to his need to access. The desired system first identifies the illegal file access request tracks for any data file. The user is provided with unique id and it cannot be change as key delegation is acquired during key generation across various TA and DA domains. User willing to access across the different domains can be tracked as he cannot change the secret id provided and being audited after regular intervals. If so found an abuse is identified and notified to the specific domain authority.
6) *Data format:* The data set transparency is provided universally to the user as it is based on cpabe-toolkit which can be used on any data type. The UAABE algorithm is used on various data types, data file formats
.txt, .doc, .docx, various audio and video formats. The same formats are checked for the encryption policy.

C. Overture
a) Bilinear Mapping: Pairing-based cryptography is the use of a pairing between elements of two cryptographic groups to a third group with a mapping e: Gi×Gj → Gk to construct or analyze cryptographic systems.
Definition: Let Gi, Gj be two cyclic groups of prime order q. A pairing is a map: e: Gi × Gj → Gk, which satisfies the following properties:
The algorithm selects a bilinear group G of prime order p with generator g and then chooses random exponents α, βi Zp , ∀i(1, 2) . To support key structure of depth d, i will range from 1 to d.
This algorithm sets the public key and master key as follows:
1) Bilinearity: ∀a, b F ∗, ∀P, Q Gi: e (P a, Qb) ≡e (P, Q) a bi) PK=(G,g,h1=gβ1

f1=g β1 ,h2 =gβ2

,f2=g β2 ,e(g, g))
2) Non-degeneracy: e (P, Q) = 1
3) Computability: there exist an efficient algorithm to compute e.
b) Access Structure: Tress access structure is used where the leaf nodes indicate attributes and nonleaf nodes are threshold gates. For any node j few of the terms considered as in [4] are:
1) numj : Number of children
2) kj : Threshold value for node j
3) Tj : Access structure for node j

Access structure plays an important role to impart security aspect. The file encryption policy and its access is managed with accordance to multiple parties requesting a resource. The DAi or an (DO/DC) i which are subgroups of participants are only allowed to join the file sharing.

D. Mathematical Model

In this paper we specify data users present in the system for uploading the data and downloading the data. The input and their respective outcome is described below in form of set theory.

1) Trusted Authority

TA = {T A}

a) TA is the root authority responsible for managing top level DA login.

b) DA maintenance and user accountability analysis.

2) Domain Authority

DA = {DA1, DA2, ...}

a) DA is the domain authority responsible for managing users.

b) DO and DC maintenance and sub accountability analysis.

3) Identify the Users

U = {u1, u2, u3,…}

a) Where 'U' may be the DO or a DC is main set of Users like u1, u2, u3...

4) Data (File) uploaded or downloaded associated with user.

D = {F 1, F 2, F 3, ...}

a) Where 'D' is the data file uploaded.

5) SYS = {UAABE-S, UAABE-KG, UAABE-E, UAABE- D}

ii) M K0 =(β1,β2,gα)

b) UAABE-KG = Pk and Mk used to generate master key (M Ki) and private key for new DA or DO(/DC) key structure. A DA is associated with a unique ID and a recursive attribute set A´ , where,

A´ = {A0 , A1 , ..., Am }

Ai = {ai,1 , ai,2 , ..., ai,n }

ai,j = jth attribute in Ai

ni = number of attributes in Ai

i) Input: PK, MK and Attribute Set

ii) Operation: CPABE-keygen (Pairing Based Cryptography)

iii) Output: M Ki - Master Key or SKdo/dc or M Ki+1

After getting the master key, DAi can authorize the next level domain authorities or users in its domain.

c) UAABE-E = To shield the data uploaded to cloud data owner encrypts the data before uploading. File is encrypted using SK under tree access structure as in [4].

For each nonroot node x several functions are defined to handle the access structure like : parent(x) : parent node of x

index(x) : index number of x

att(x) : if x is leaf node and denotes attributes associated with leaf node x in the tree.

i) Input: SK, File and Tree Access Structure (User defined)

ii) Operation: CPABE-enc (analogous to access tree policy)

iii) Output: Encrypted File.

Before encryption process the DO defines a tree access structure T for an file that is to be processed.

d) UAABE-D = User SK is used for decryption of the file. Decryption algorithm verifies key structure with SKu with the tree access structure T associated with CT. If

A´ does not satisfy T

a) UAABE-S = Initial System setup which generates the public key (Pk) and master key (Mk).

algorithm will return 0, else selects i from set returned by T (A´). Then node is decrypted using i and att(t) where t is node from T . The process is recursively done across the access structure from node to root R

i) Input: User Secret Key, PK and Encrypted File

ii) Operation: CPABE-dec iii) Output: File

Across the entire system defined above at each level of hierarchy and track is maintained at the access structure defined by user. Analyzing the data and access policy of user leads to some range towards maintaining user accountability.

E. Multiplexor Logic

The proposed system takes number of inputs and has various outputs according to states. These outputs can be represented by using a simple multiplexor logic. As shown in Fig. 2 number of active environments remain in system according to active entities like TA, DA, DO and DC.

1

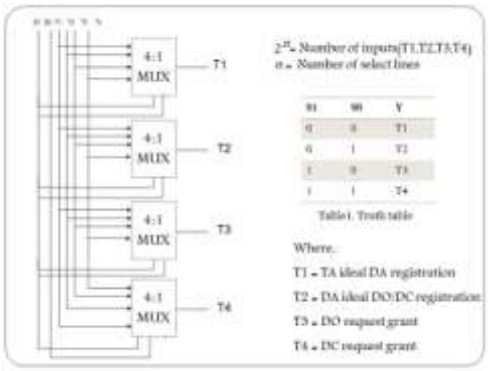

Fig. 2: Role played by CSP

Fig2. is an multiplexor logic for initial system setup and has the function of granting and uploading the file over the Cloud.

F. Operating Environment

An UAABE toolkit based on the cpabe toolkit (http://acsc.csl.sri.com/cpabe/) developed for CP-ABE [3] which employ the Pairing-Based Cryptography library (http://crypto.stanford.edu/pbc/) is implemented. The tree access structure used across the hierarchy. DO=DC are online as an when they have to access the file. CSP and TA are kept always online to manage the system request/response. Various extensive experiments are conducted on a system with dual core 2.30-GHz CPU and 8-GB RAM, running Windows 8 as host and Ubuntu Server 10.04 for Ubuntu Enterprise Cloud setup. Experimental data is analyzed and statistical data is given.

IV. RESULTS AND DISCUSSION

As per number of entities before considering file upload and download hierarchical validator is defined. As shown in Fig.
3 the new registration of DO or DO or DC is done across the system by validating the top authority only followed by preceding validators. The user is provided with unique id and

Fig. 3: Key Delegation: Client Side

it cannot be change as key delegation is acquired during key generation across various TA and DA domains. User willing to access across the different domains can be tracked as he cannot change the secret id provided and being audited after regular intervals.
The file is uploaded on to the CSP by the DO and downloaded by the DC. During the upload process at the CSP side various operations are performed according to the details provided by the data handler. If it is an initial login for an data upload then as shown in Fig. 4 initially an private key is generated for the user using the master key, public key and various attributes as specified by user.

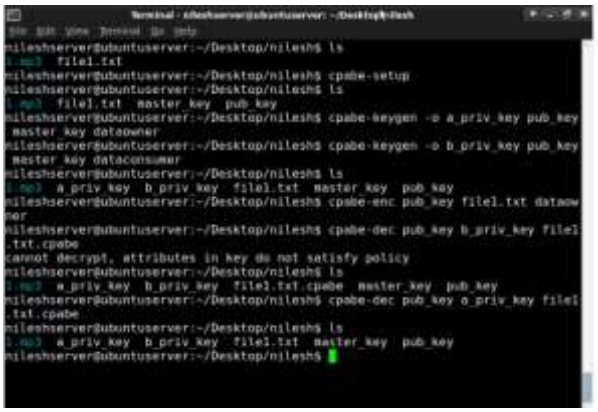

Fig. 4: Role played by CSP

The file is encrypted using the public key and various attributes of the file. The encrypted file can be decrypted and downloaded at the user side if its attributes match with the attributes used for encryption. If the attributes do not match then the error will be displayed for unsatisfiablity of attribute The Fig. 5 shows the same process of key generation for two

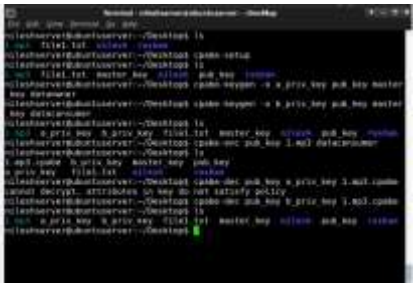

Fig. 5: Role played by CSP

users using two different attributes and encryption for a .mp3 format.

Also as an access policy error is generated which can be maintained by for tracking the users and analysing it for find- ing out users involved in key abuse. The user accountability at other level of DA login can be identified by analysing the same by maintaining the login and data access log at each level of hierarchy.

V. CONCLUSION

After summarizing work, in this model secure outsourcing of data and arbitrary computations thereon

consisting of DO managed by TA/DA according to hierarchical policy. The CSP is mostly involved in evaluation under encryption and in parallel by the untrusted Commodity Cloud. The given instantiation of model is based on HABE scheme. The scheme not only provides secure access control, but achieves efficient user accountability by maintaining and analyzing the data at each hierarchy level according to the system requirement. In future this state of art can be further extended by maintaining hierarchical networking data to analyze and move a step forward towards identifying the specific key abuse. Also a algorithmic modifications in general access control policies can be focused to keep vigorous improvement towards security and user accountability.

ACKNOWLEDGMENT

I take this opportunity to express my profound gratitude and deep regards to my guide Ms. D. A. Chaudhari for her exemplary guidance, monitoring and constant encouragement throughout the course of this project. I also take this opportunity to express a deep sense of gratitude to my Head of the department Mrs. M. A. Potey, PG Coordinator Mrs. S. S. Pawar, for her cordial support, valuable information and guidance. Thanks to all those who helped me in completion of this work knowingly or unknowingly like all those researchers, my lecturers and friends.

REFERENCES

[1] Jian Chang, Krishna K. Venkatasubramanian, Member, IEEE, Andrew G. West, Sampath Kannan, Insup Lee, Fellow, IEEE, Boon Thau Loo, and Oleg Sokolsky, Member, IEEE, "AS-CRED: Reputation and Alert Service for Inter domain Routing", IEEE SYSTEMS JOURNAL, VOL. 7, NO. 3, SEPTEMBER 2013

[2] J. Chang, K. Venkatasubramanian, A. G. West, S. Kannan, B. T. Loo, O.Sokolsky, and I. Lee, "AS-TRUST: A trust characterization scheme for autonomous systems BGP," in Proc. 4th Int. Conf. TRUST, Jun. 2011, pp. 262276.

[3] H. Kim and J. Huh, "Detecting DNS-poisoning-based phishing attacks from their network performance characteristics," Electron. Let. vol. 47, no. 11, pp. 656658, 2011.

[4] S. Abu-Nimeh and S. Nair, "Circumventing security toolbars and phishing filters via rogue wireless access points," Wireless Commun. Mobile Compute. vol. 10, no. 8, pp. 11281139, 2010.

[5] A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers," SIGCOMM Compute. Commun. Rev., vol. 36, no. 4, pp. 291302, 2006.

[6] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP mis configuration," in Proc. Conf. Appl., Technol., Arch. Protocols Compute. Commun. 2002, pp. 316.

[7] V. J. Bono. (1997, Apr. 26). 7007 Explanation and Apology [Online]. Available: http://www.merit.edu/mail.archives/nanog/1997-l

[8] S. Kent, C. Lynn, and K. Seo, "Secure border gateway protocol (SBGP)," IEEE J. Selected Areas Commun., vol. 18, no. 4, pp. 582592, Apr. 2000.

[9] J. Ng. (2002, Oct.). Extensions to BGP to Support Secure Origin BGP (soBGP) [Online]. Available: http://tools.ietf.org/html/draft-ng-sobgpbgp- extensions-00.

[10] M. Lepinski and S. Turner. (2012, May 8). An Overview of BGPSEC [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-sidr-bgpsec-overview/.

[11] X. Hu and Z. M. Mao, Accurate real-time identification of IP prefix hijacking, in SP: Proc. IEEE Symp. Security Privacy, May 2007, pp. 317 .

[12] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, iSPY: Detecting IP prefix hijacking on my own, SIGCOMM Compute. Commun. Rev., vol. 38, no. 4, pp. 327338, 2008.

[13] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, PHAS: A prefix hijack alert system, in Proc. 15th Conf. USENIX Security Symp., vol. 15. 2006, article 11.

[14] J. Karlin, S. Forrest, and J. Rexford, Autonomous security for au- tonomous systems, Compute. Newt. vol. 52, no. 15, pp. 29082923, 2008.

Nilesh S Jadhav received the B.E. degree in Com- puter Sciences from Siddhant College of Engineer- ing Sudumbare, Pune in 2009. During 2010-2012, he did lecturership in Siddhant College of Engi- neering,Pune and 2012-2013 in D.Y.Patil College of Engineering Akurdi,Pune. Now he is pursuing Mas- ter degree in Computer Engineering from D.Y.Patil College of Engineering Akurdi,Pune.

Dipalee A. Chaudhari received the BE degree in Computer Science and Engineering from University of Pune in 2000 and ME in Computer Engineering from University of Pune in 2010 and has 8 years of teaching experience. She is currently working as Assistant professor at D. Y. Patil College of EngineeringAkurdi, Pune.