# A Cross Tenant Access Control (CTAC) Model for Cloud Computing: Formal Specification and Verification

M Venkatesu[1], Dr.G Anjan Babu[2]
[1]*Student,Dept.of MCA,SV University college of CM&CS*
[2]*Professor,Dept.ofMCA,SVUniversitycampusofCM&CS,Tirupati*

***Abstract*- Sharing of resources on the cloud can be achieved on a large scale since it is cost effective and location independent. Despite the hype surrounding cloud computing, organizations are still reluctant to deploy their businesses in the cloud computing environment due to concerns in secure resource sharing. In this paper, we propose a cloud resource mediation service offered by cloud service providers, which plays the role of trusted third party among its different tenants. This paper formally specifies the resource sharing mechanism between two different tenants in the presence of our proposed cloud resource mediation service. The correctness of permission activation and delegation mechanism among different tenants using four distinct algorithms (Activation, Delegation, Forward Revocation and Backward Revocation) is also demonstrated using formal verification. The performance analysis suggest that sharing of resources can be performed securely and efficiently across different tenants of the cloud.**

## I. INTRODUCTION

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.

Structure of cloud computing

How Cloud Computing Works: The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games.

The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing.

Characteristics and Services Models The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below:

- On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.
- Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
- Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction

(e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

- Rapid elasticity: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.
- Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service.
- Services Model: Cloud Computing comprises three different service models, namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The three service models or layer are completed by an end user layer that encapsulates the end user perspective on cloud services. The model is shown in figure below. If a cloud user accesses services on the infrastructure layer, for instance, she can run her own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications herself. If she accesses a service on the application layer, these tasks are normally taken care of by the cloud service provider.

Benefits of cloud computing:

1. Achieve economies of scale – increase volume output or productivity with fewer people. Your cost per unit, project or product plummets.
2. Reduce spending on technology infrastructure. Maintain easy access to your information with minimal upfront spending. Pay as you go (weekly, quarterly or yearly), based on demand.
3. Globalize your workforce on the cheap. People worldwide can access the cloud, provided they have an Internet connection.
4. Streamline processes. Get more work done in less time with less people.
5. Reduce capital costs. There's no need to spend big money on hardware, software or licensing fees.
6. Improve accessibility. You have access anytime, anywhere, making your life so much easier!
7. Monitor projects more effectively. Stay within budget and ahead of completion cycle times.
8. Less personnel training is needed. It takes fewer people to do more work on a cloud, with a minimal learning curve on hardware and software issues.
9. Minimize licensing new software. Stretch and grow without the need to buy expensive software licenses or programs.
10. Improve flexibility. You can change direction without serious "people" or "financial" issues at stake.

Advantages:

1. Price: Pay for only the resources used.
2. Security: Cloud instances are isolated in the network from other instances for improved security.
3. Performance: Instances can be added instantly for improved performance. Clients have access to the total resources of the Cloud's core hardware.
4. Scalability: Auto-deploy cloud instances when needed.
5. Uptime: Uses multiple servers for maximum redundancies. In case of server failure, instances can be automatically created on another server.
6. Control: Able to login from any location. Server snapshot and a software library lets you deploy custom instances.
7. Traffic**:** Deals with spike in traffic with quick deployment of additional instances to handle the load.

## II IMPLEMENTATION:

MODULES:

- ❖ System Framework
- ❖ The Responsibilities of Entities
- ❖ Steps Involved for Initiating a Permission Activation Request
- ❖ Revocation

MODULES DESCRIPTION:

System Framework:

In this paper formally specifies the resource sharing mechanism between two different tenants in the presence of our proposed cloud resource mediation service. There are three main entities. To explain the service, we use an example involving two tenants, T1 and T2, where T1 is the Service Provider (SP) and T2 is the Service Requester(SR) (i.e. user) and the CRMS. T1 must own some permission pi for which user of T2 can generate a cross-tenant request. The resource request from a user of T2 must be submitted to T1, which then handovers the request to the CRMS for authentication and authorization decisions. The CRMS evaluates the requestbased on the security polices provided by T1.

The Responsibilities of Entities:

a) Tenant T1 responsibilities: T1 is responsible for publishing cross tenant policies on the CRMS. T1 receives access requests from T2 and redirects the request to the CRMS for further processing.

b) Tenant T2 responsibilities: The CRMS redirects access requests to T2 for authentication. Once the redirectedaccess request is received, the responsibility of T2 is to authenticate the identity of particular user. In response, T2 sends the user authentication response (valid or invalid) and tenant authentication response to the CRMS.

c) CRMS responsibilities: The CRMS receives the permission-activation request redirected from T1. Once an access request is received, the CRMS evaluates the request on the pre-published policies and responds to T1.

Steps Involved For Initiating a Permission Activation Request:

Step 1: Permission activation request: A user wishing to access a resource at T1. The user will be presented a directory where a list of shared services along with their descriptions are present.

Step 2: Request redirection to the CRMS: Upon selectionof a shared service the user wishes to access, the user is redirected to the CRMS site. On the site, the user will be asked for the parent tenant. The user selects the parent tenant and the CRMS redirects the user's request to the selected tenant (T2 in this case).

Step 3: Tenant T2 authentication: The user has to authenticate at her parent tenant, T2. Upon successful authentication, the user will be redirected again to CRMS with the attributes requested by the CRMS for cross tenant policy execution.

Step 4: CRMS redirection to tenant T1 & permission activation: The user's attributes are evaluated against the T1 policy and if the policy criteria is successfully fulfilled, then the user is provided service access at T1; otherwise, the access request is denied. The CRMS also takes into account any conflict of interest policies, such as Chinese Wall Policy.

Revocation:

There are two ways in which we can revoke a previously granted permission from the cross-tenant user/cross-tenant. To achieve the permission revocation, we introduce the Forward Revocation Algorithm and the Backward Revocation Algorithm .A forward revocation query defines a request in which an intra-tenant user revoke a permission or a set of permissions from a cross tenant user/tenant along with the deactivation of the delegation policy. And A Backward revocation query defines a action that is triggered when the attributes of the delegatee mismatch. Thus, an intra-tenant userrevokes a permission or a set of permissions from a crosstenant user/tenant as well as deactivating the delegation policy.

## III INPUT DESIGN AND OUTPUT DESIGN

INPUT DESIGN:

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.

- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

## IV SYSTEM_STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

## V LITERATURE SURVEY

1) DaSCE: Data Security for Cloud Environment with Semi-Trusted Third Party

AUTHORS: Ali, M., Malik, S. and Khan, S.,

Off-site data storage is an application of cloud that relieves the customers from focusing on data storage system. However, outsourcing data to a third-party administrative control entails serious security concerns. Data leakage may occur due to attacks by other users and machines in the cloud. Wholesale of data by cloud service provider is yet another problem that is faced in the cloud environment. Consequently, high-level of security measures is required. In this paper, we propose Data Security for Cloud Environment with Semi-Trusted Third Party (DaSCE), a data security system that provides (a) key management (b) access control, and (c) file assured deletion. The DaSCE utilizes Shamir's (k, n) threshold scheme to manage the keys, where k out of n shares are required to generate the key. We use multiple key managers, each hosting one share of key. Multiple key managers avoid single point of failure for the cryptographic keys. We (a) implement a working prototype of DaSCE and evaluate its performance based on the time consumed during various operations, (b) formally model and analyze the working of DaSCE using High Level Petri nets (HLPN), and (c) verify the working of DaSCE using Satisfiability Modulo Theories Library (SMT-Lib) and Z3 solver. The results reveal that DaSCE can be effectively used for security of outsourced data by employing key management, access control, and file assured deletion.

2) Control Cloud Data Access Privilege and Anonymity With Fully Anonymous Attribute-Based Encryption

AUTHORS: Jung, T., Li, X. Y., Wan, Z. and Wan,M

Cloud computing is a revolutionary computing paradigm which enables flexible, on-demand and low-cost usage of computing resources, but the data is outsourced to some cloud servers, and various privacy concerns emerge from it. Various schemes based on the Attribute-Based Encryption have been proposed to secure the cloud storage. However, most work focuses on the data contents privacy and the access control, while less attention is paid to the privilege control and the identity privacy. In this paper, we present a semi-anonymous privilege control scheme AnonyControl to address not only the data privacy but also the user identity privacy in existing access control schemes. AnonyControl decentralizes the central authority to limit the identity leakage and thus achieves semi-anonymity. Besides, it also generalizes the file access control to the privilege control, by which privileges of all operations on the cloud data can be managed in a fine-grained manner. Subsequently, we present the AnonyControlF which fully prevents the identity leakage and achieve the full anonymity. Our security analysis shows that both Anonymous Control and Anonymous Control-F are secure under the DBDH assumption, and our performance evaluation exhibits the feasibility of our schemes.

3) Fine-Grained Two-Factor Access Control for Web-Based Cloud Computing Services

AUTHORS**:** Liu, J. K., Au, M. H., Huang, X., Lu, R., and Li, J

In this paper, we introduce a new fine-grained two-factor authentication (2FA) access control system for web-based cloud computing services. Specifically, in our proposed 2FA access control system, an attribute-based access control mechanism is implemented with the necessity of both a user secret key and a lightweight security device. As a user cannot access the system if they do not hold both, the mechanism can enhance the security of the system, especially in those scenarios where many users share the same computer for web-based cloud services. In addition, attribute-based control in the system also enables the cloud server to restrict the access to those users with the same set of attributes while preserving user privacy, i.e., the cloud server only knows that the user fulfills the required predicate, but has no idea on the exact identity of the user. Finally, we also carry

out a simulation to demonstrate the practicability of our proposed 2FA system.

4) Jobber: Automating inter-tenant trust in the cloud

AUTHORS: Sayler, A., Keller, E. and Grunwald, D

Today, a growing number of users are opting to move their systems and services from self-hosted data centers to cloud-hosted IaaS offerings. These users wish to both benefit from the efficiencies that shared multitenant hosting can offer while still retaining or improving the kinds of security and control afforded by self-hosted solutions. In this paper, we present Jobber: a highly autonomous multi-tenant network security framework designed to handle both the dynamic nature of cloud datacenters and the desire for optimized inter-tenant communication. Our Jobber prototype leverages principals from Software Defined Networking and Introduction Based Routing to build an inter-tenant network policy solution capable of automatically allowing optimized communication between trusted tenants while also blocking or rerouting traffic from untrusted tenants. Jobber is capable of automatically responding to the frequent changes in virtualized data center topologies and, unlike traditional security solutions, requires minimal manual configuration, cutting down on configuration errors.

5) Cross-tenant trust models in cloud computing

AUTHORS: Tang, B. and Sandhu, R.

Most cloud services are built with multi-tenancy which enables data and configuration segregation upon shared infrastructures. Each tenant essentially operates in an individual silo without interacting with other tenants. As cloud computing evolves we anticipate there will be increased need for tenants to collaborate across tenant boundaries. This will require cross-tenant trust models supported and enforced by the cloud service provider. Considering the on-demand self-service feature intrinsic to cloud computing, we propose a formal cross-tenant trust model (CTTM) and its role-based extension (RB-CTTM) integrating various types of trust relations into cross-tenant access control models which can be enforced by the multi-tenant authorization as a service (MTAaaS) platform in the cloud.

## VI SYSTEM ANALYSIS

EXISTING SYSTEM:

- Zhao et al. propose a cross-domain single sign on authentication protocol for cloud users, whose security was also proven mathematically. In the approach, the CSP is responsible for verifying the user's identity and making access control decisions.
- As computing resources are being shared between tenants and used in an on-demand manner, both known and zeroday system security vulnerabilities could be exploited by the attackers (e.g. using side-channel and timing attacks).
- In existing, a fine grained data-level access control model (FDACM) designed to provide role-based and data-based access control for multi-tenant applications was presented. Relatively lightweight expressions were used to represent complex policy rules.

DISADVANTAGES OF EXISTING SYSTEM:

- Traditional access control models, such as role based access control, are generally unable to adequately deal with cross-tenant resource access requests.
- Specification level security is difficult to achieve at the user and provider ends.
- The security of the approach was not provided.

PROPOSED SYSTEM:

- We use model checking to exhaustively explore the system and verify the finite state concurrent systems. Specifically, we use High Level Petri Nets (HLPN) and Z language for the modeling and analysis of the CTAC model.

## VII SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

*TYPES OF TESTS*

*Unit testing*

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

*Functional test*

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures:interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

*System Test*

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

*White Box Testing*

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

*Black Box Testing*

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

## VIII CONCLUSION

In this paper, we proposed a cross-tenant cloud resource mediation service (CRMS), which can act as a trusted-third party for fine-grained access control in a cross-tenant environment. For example, users who belong to an intra-tenant cloud can allow other cross-tenant users to activate permission in their tenant via the CRMS. We also presented a formal mode lCTAC with four algorithms designed to handle the requests for permission activation. We then modeled the algorithms using HLPN, formally analyzed these algorithms in Z language ,and verified them using Z3 Theorem Proving Solver. The results obtained after executing the solver demonstrated that the asserted algorithm specific access control properties were satisfied and allows secure execution of permission activation on the cloud via the CRMS.

Future work will include a comparative analysis of the proposed CTAC model with other state-of-the-art cross domain access control protocols using real-world evaluations. For example ,one could implement the protocols in a closed or smallscale environment, such as a department within a university.This would allow the researchers to evaluate the performance, and potentially (in)security, of the various approaches underdifferent real-world settings.

## REFERENCES

[1] Akhunzada, A., Gani, A., Anuar, N. B., Abdelaziz, A., Khan, M. K., Hayat, A., & Khan, S. U. (2016). Secure and dependable software defined networks. Journal of Network and Computer Applications, 61, 199-221.

[2] Alam, Q., Tabbasum, S., Malik, S., Alam, M., Tanveer, T., Akhunzada, A., Khan, S., Vasilakos, A. and Buyya, R., (2016). Formal Verification of the xDAuth Protocol. IEEE Transactions on Information Forensics and Security, 11(9), pp. 1956-1969.

[3] Ali, M., Malik, S. and Khan, S., DaSCE: Data Security for Cloud Environment with Semi-Trusted Third Party.

[4] Barrett, C., Conway, C.L., Deters, M., Hadarean, L., Jovanovi, D., King, T., Reynolds, A. and Tinelli, C., 2011, July. Cvc4. In International Conference on Computer Aided Verification (pp. 171-177). Springer Berlin Heidelberg.

[5] Bofill, M., Nieuwenhuis, R., Oliveras, A., Rodrguez-Carbonell, E. andRubio, A., 2008, July. The barcelogic SMT solver. In International Conference on Computer Aided Verification (pp. 294-298). Springer Berlin Heidelberg.

[6] Bruttomesso, R., Cimatti, A., Franzn, A., Griggio, A. and Sebastiani, R.,2008, July. The mathsat 4 smt solver. In International Conference on Computer Aided Verification (pp. 299-303). Springer Berlin Heidelberg.

[7] Choo, K.K., 2006. Refuting security proofs for tripartite key exchange with model checker in planning problem setting. In 19th IEEE Computer Security Foundations Workshop (CSFW'06) (pp. 12-pp).IEEE.

[8] Choo, K.-K. R., Domingo-Ferrer, J. and Zhang, L., 2016. Cloud Cryptography: Theory, Practice and Future Research Directions. Future Generation Computer Systems, 62, pp. 51-53.

[9] De Moura, L. and Bjørner, N., 2011. Satisfiability modulo theories: introduction and applications. Communications of the ACM, 54(9), pp.69- 77.

[10] Dutertre, B. and De Moura, L., 2006. The yices smt solver.Tool paperat http://yices.csl. sri. com/tool-paper. pdf, 2(2).

[11] Heiser, J., 2009. What you need to know about cloud computing security and compliance. Gartner, Research, ID, (G00168345).

[12] Jung, T., Li, X. Y., Wan, Z. and Wan, M., 2015. Control Cloud Data Access Privilege and Anonymity WithFully Anonymous Attribute-Based Encryption. IEEE Transactions on Information Forensics and Security, 10(1), (pp. 190-199).

[13] Lin, Y., Malik, S.U., Bilal, K., Yang, Q., Wang, Y. and Khan, S.U., 2016. Designing and Modeling of Covert Channels in Operating Systems.IEEE Transactions on Computers, 65(6), pp.1706-1719.

[14] Liu, J. K., Au, M. H., Huang, X., Lu, R., and Li, J., 2016. Fine-Grained Two-Factor Access Control for Web-Based Cloud Computing Services.IEEE Transactions on Information Forensics and Security, 11(3), (pp. 484-497).

[15] Liu, X., Deng, R. H., Choo, K.-K. R. and Weng, J., 2016. An EfficientPrivacy-Preserving Outsourced Calculation Toolkit With Multiple Keys. IEEE Transactions on Information Forensics and Security, 11(11), pp. 2401-2414.

[16] Ma, K., Zhang, W. and Tang, Z., 2014. Toward Fine-grained Data-levelAccess Control Model for Multi-tenant Applications.International Journalof Grid and Distributed Computing, 7(2), pp.79-88.

[17] Murata, T., 1989. Petri nets: Properties, analysis and applications. Proceedings of the IEEE, 77(4), pp.541-580.

[18] Sayler, A., Keller, E. and Grunwald, D., 2013. Jobber: Automating inter-tenant trust in the cloud. In Presented as part of the 5th USENIX Workshop on Hot Topics in Cloud Computing.

[19] SMT-Lib. http://smtlib.cs.uiowa.edu/, 2015.

[20] Tang, B. and Sandhu, R., 2013, August. Cross-tenant trust models in cloud computing. In Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on (pp. 129-136). IEEE.

[21] Yang, Y., Zhu, H., Lu, H., Weng, J., Zhang, Y. and Choo, K.-K. R., 2016. Cloud based data sharing with fine-grained proxy re-encryption. Pervasive and Mobile Computing, 28, pp. 122-134.

[22] Zhang, Y., Patwa, F., Sandhu, R. and Tang, B., 2015, August. Hierarchical secure information and resource sharing in openstack community cloud. In Information Reuse and Integration (IRI), 2015 IEEE International Conference on (pp. 419-426). IEEE.

[23] Zhao, G., Ba, Z., Wang, X., Zhang, Y., Huang, C. and Tang Y., 2016. Constructing Authentication Web in Cloud Computing. Security and Communication Networks, 9(15), pp. 2843-2860