Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 4, April 2015, pg.65 – 74

RESEARCH ARTICLE

PRIVACY MONITORING IN THE ONLINE SYSTEM USING COLLABORATIVE STRATEGY MANAGEMENT

Sridevi.S[1], **M.Tech, Indumathi.N**[2], **M.Tech**
Research Scholar, Department of Computer Science and Engineering
SRM University, Chennai, Tamil Nadu, India
Email: sridevishiva76@gmail.com [1], induarivu27@gmail.com [2]

***Abstract— Collaborative strategy management is one of the efficient methods of guiding principle supervision in order to protect perceptive data loss. In Collaborative strategy management, can refer to other similar policies to set up their own policies to protect privacy and other susceptible information. In this work an Improved Collaborative strategy management is proposed and being evaluated for more effective application to ensure improved security in many networking applications.** However, the overclaim of privileges is widespread in emerging applications, including mobile applications, and social network services, because the applications' users involved in policy management have little knowledge of policy-based management. The overclaim can be leveraged by malicious applications, then lead to serious privacy leakages and financial loss. To resolve this issue, this paper proposes a novel policy management mechanism, referred to as collaborative strategy management (CSM for short). Furthermore, to obtain similar policies more effectively, which is the key step of CSM, a text mining-based similarity measure method is presented. We evaluate CSM with the data of Android applications and demonstrate that the text mining-based similarity measure method is more effective in obtaining similar policies than the previous category-based method.*

Keywords- Improved Collaborative strategy management, susceptible, Verification, text mining, android application, privacy leakage

I. INTRODUCTION

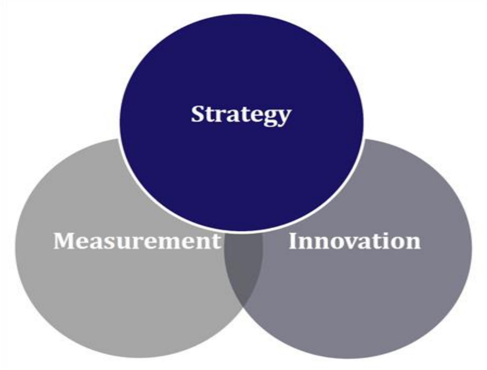
All every software application domain should provide and make sure security issues. For the most part, as countries around the world transition from paper-based to electronic information record infrastructures, compliance with these data protection laws will require sophisticated information management technologies. Technical and policy challenges in relation to the widespread adoption of electronic information records systems have been discussed. There are also different aspects between the users and the service providers. Often organizations struggle to bring their strategies into action. Management teams don't always agree on the vision, strategies, are vague and short of differentiation, organizations are paralyzed by lack of initiatives or inconsistent behavior, different organizational entities work in different directions, people are disconnected and lack engagement, organizations consistently underperform.

Sustainable methodologies to address these issues' do exist. Introducing the principle of the strategy focused organization provides a platform for optimizing and integrating the building blocks that constitute an effective management system. There is no single solution that fits to all organizations. The basic principles of the strategy management of the organization is

- ✓ Mobilize change through executive leadership
- ✓ Translate strategy to operational terms
- ✓ Align the organization to the strategy

- ✓ Motivate to make strategy everyone's job
- ✓ Govern to make strategy a continual process

Participation and strategic dialogue are key characteristics of our approach to align strategy formulation and execution. By involving people, exchanging information and fostering strategic learning, capabilities for strategy execution are development throughout the organization.



The majority of users want to disclosure only least privacy data, and the service providers request at most personal information. Under this situation, if most right of information management comes up to the service providers, it provides the unfair position to the users. It is the drawback of monopolistic information management technologies. Security issues usually are derived from laws such as data protection acts or general security rules branching from the domain itself. The policy-based management is extensively used technique to deal with complex and large-scale network systems. Traditionally, construction of policy-based management consists of four core components as in Fig 1: policy enforcement point (PE), policy decision point (PD), policy repository (PR), policy administration point (PA) as shown in figure. The policies in PA are specified and verified by policy administrator or group and also the policies in PR are deployed by them. Once the system runs, the applicable policies from PR will be retrieved by PD and conclusion will be made. In case the subject wants to open a file (authorization action) or launch a logger to record system context (obligation action), PE takes control of the decisions.

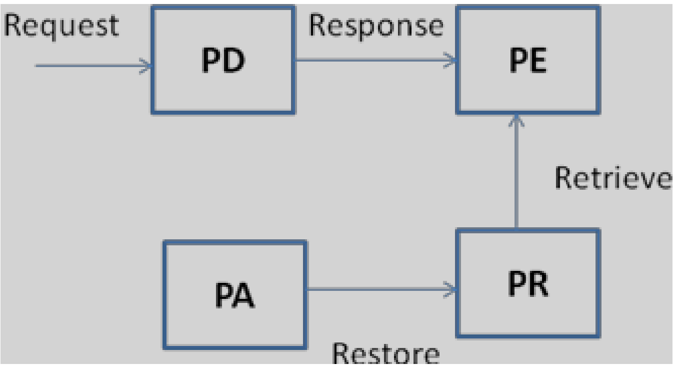
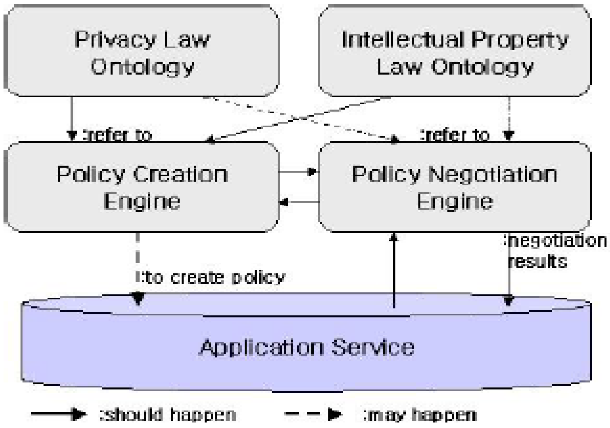


Fig 1: Policy Driven Management Architecture

In Collaborative Privacy Management System (Co- PMS) each user provides own privacy policy by using a policy creation interface, as in Fig.2 This interface is useful in making the specific privacy policy ontology for the each user. Each individual stores the encrypted privacy policy in his/her potable storages or in his/her computer. And then they can use it as new authentication certification. The policy negotiation engine performs collaborative process with the certain service's data disclosure policy.

Fig. 2 Policy Based Privacy Management System Architecture



In case of application of the policy-based management method to emerging applications such as mobile and social network services, if more privileges than those are normally essential by a subject are assigned by not well-trained administrator, cause real troubles. In Android application development, three responsibilities are usually involved in the policy administration: Application Developers declare the permissions requested by the application; Application Marketers verify if the application is lawful; Application Users decide about approval of the consent.

II. RELATED WORK

Policy as follows: "A policy is a set of rules reflecting an overall strategy or objective, affecting the behavior of agents and thus designed to help control and administer a system". Initially an agent-based architecture, which encompasses several agents that work together to provide the Policy Management services to the applications. In this architecture, and the Policy Service Agent and the Policy Management Agent are two agents of particular interest.

- The role of Policy Management Agent is defining, editing, storing and assigning policies. To get done this task, the Policy Management Agent may access the application profile accumulated in the Policy Information Base.
- The responsibility of the Policy Service Agent is to carry out the task of interpreting and enforcing policies. This requires a continuous communication between the application and the policy service agent.
- This communication is utilized to negotiate and exchange policy information updates that would impact the behavior of the agents representing the application at the run-time.

Traditional policy system has following demerits:

The marketers on average tend to permit more applications regardless of the malicious permission requests. The application users may not know what the demanded permissions mean, thus approving all requests because they are eager to use the application.

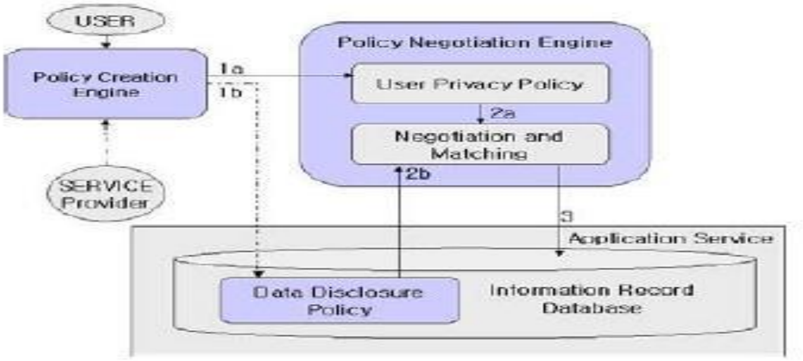
Co-PMS Architecture

The entire Co-PMS is comprised of three stages policy creation stage, policy negotiation stage and lastly application service retrieval as in Fig.3.

- *Policy Creation Stage*: In the policy creation stage Fig.3, there are two components of the policy - the privacy policy and data disclosure policy. Service providers create the data disclosure policy through the policy creation engine. The policy manages the access privileges for each role according to the category of information feature, the purpose of request, and the projected recipient of results. Using the policy creation engine user creates the his/her own privacy policy. This encrypted policy controls the leakage of his/her personal information, also according to the category of information feature, the intended recipient, and the purpose of the request.

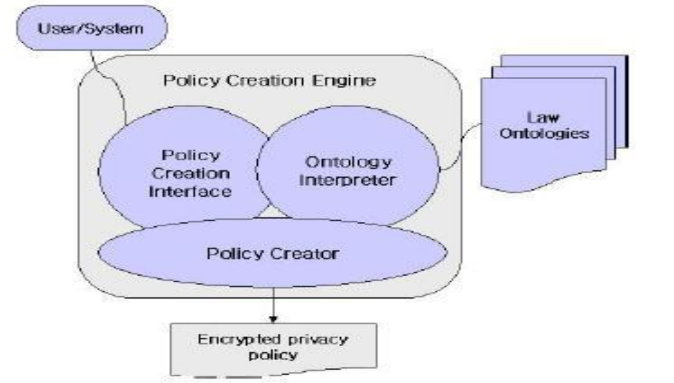
- The policy creation engine has three main elements such as a policy creation interface, an ontology interpreter, and a policy creator. With the help of policy creation interface users were easily define their privacy policy without the expertise. The ontology interpreter imports the privacy law ontology and the intellectual property law ontology. After integration of the rules from ontologies policy creator provides encrypted privacy policy.

Fig 3: Collaborative Privacy Management System Architecture



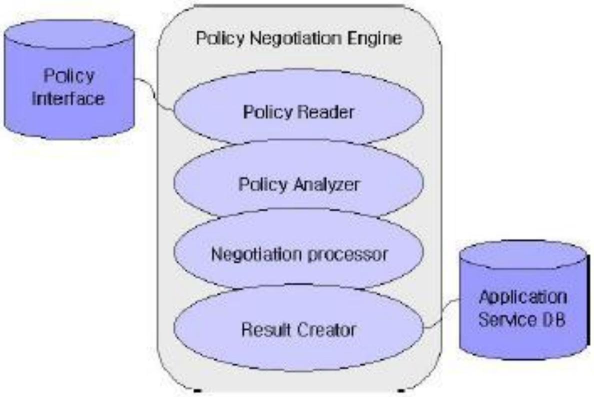
- *Policy Negotiation Stage :* In the policy negotiation stage In Fig.3, the user is informed to the system organization‘s policies concerning data use and disclosure, advised of any disagreement with one’s own privacy and security preferences. This fully automated process is completed before the user provides any personal data to the organization. The user first uses the policy creation engine to convey his personal policy concerning the use and disclosure of his personal data. This information is matched with the system organization’s privacy and security policies to identify any conflicts. The user gives the suggestion of these conflicts and given a chance to resolve them or terminate the process.

Fig 4: Policy Creation Engine



The user should update the policy regarding whether his data may be disclosed to third parties or utilized for a different purpose than for which it was accumulated. This modified information are recorded as a result of policy negotiation in the application service database. It is factored in at the time of service processing. The policy negotiation engine has mainly four elements such as a policy reader, a policy analyzer, negotiation processor, and result creator, as in Fig.5.

Fig 5: Policy negotiation engine



The reader imports the privacy policy and the data disclosure policy. The analyzer matches for each entry of the data disclosure policy. The analyzer matches for each entry of the policies.

- *Application Service Stage* : In the application service retrieval stage in Fig.3, the application system controls accesses based upon the user's purpose, role, and intended recipient. The service system apply the result of negotiation as a kind of user's certification , user's information, and a agreement between service provider and user. This system already installed the result of policy negotiation between the privacy policy and the data disclosure policy. Through a database interface this Co-PMS can be integrated into existing environments.

III. EXISTING SYSTEM

The traditional framework of policy based management consists of four core components: PDP (Policy Decision Point), PEP (Policy Enforcement Point), PAP (Policy Administration Point) and PR (Policy Repository). A well-trained policy administrator or group will specify, verify policies in PAP, and deploy the policies in PR. After a system runs, PDP will retrieve applicable policies from PR, and make decisions. PEP takes charge of the decision, such as satisfying the request where a subject wants to open a file (authorization action), or launching a logger to record system context (obligation action). The over claim of privileges, where a not well-trained administrator assigns more privileges than those are required of a subject, is a increasingly serious problem, especially when the method of policy based management is applied to emerging application scenarios, such as mobile applications and social network services. For instance, during the process of Android application development, three roles are usually involved in the policy administration: Application Developers declare which permissions the application will request; Application Marketers verify whether the application is legitimate or not by an automatic tool; Application Users decide whether to approve the permission requests. These three roles are usually performed by those who are not well-trained in policy based management.

Disadvantages of existing system:

- ✓ The marketers usually tend to allow malicious permission requests.
- ✓ The application users may not know what the requested permissions mean, thus approving all requests because they are eager to use the application. The same issue exists in social network services, where a user is asked to grant access to private data to third-party applications. This challenge to policy administration is increasing serious due to the explosion of these applications.

IV. PROPOSED SYSTEM

In this study, the Collaborative strategy management (CSM) is presented. As the vital idea of CPA (collaborative policy administration) is that applications with related functionalities shall have similar policies, CPA will examine policies already specified by other similar applications and perform collaborative recommendation. The extent of similarity will be calculated by a text mining based algorithm. The major enhancement proposed in CSM over CPA is the investigation of safety definitions using real time environments, such as online social networking datasets. Additionally, efforts are being made to improve permission model method of CPA using fine grain access control. Also more security will be provided for real time application like social networks by the administrator.

Advantages of proposed system:

- ✓ Two main functions in policy administration are defined based on similarity measure methods, which will select similar policies as a refinement basis to assist administrators to design or verify their target policies.
- ✓ We propose a text mining based similarity measure method to help policy administrators to obtain similar policies.
- ✓ The framework supports two types of user interfaces, and provides functions of collaborative policy design and collaborative policy verification.

V. BACKGROUNG AND MOTIVATION

No trusted Administration Point A professional expert or group will take charge of the policy administration in the traditional administration model. However, upcoming applications, especially social network services mobile applications, face up to the existing trust model in the policy administration. A developer of a third party application must request the privileges to be used by the application and may not know what is at risk if an application requests privileges. This breaks the basic security principle that is principle of least privilege. Thus, the changing trust model is the novel strategy to strengthen the policy administration.

Motivated Scenario:

- o *Android Application*: In the Android security framework, a developer sets the permissions for an application requested by various users. End user decides whether the requested permissions are legal for his or her mobile device. Due to the openness of the Android security framework, hundreds of millions of developers and users are involved. Though there is no tools' support, the developers could misunderstand the description of the requested permissions. As a result, the overclaim of permissions is widespread in Android applications
- o *Social Network Services:* In social network services, third-party web-based applications could request susceptible information of end users. The end user approves sensitive requests one by one is allowed in social network services. The developer can decide which sensitive requests can be set according to other similar policies. So, he or she can develop securer and more satisfactory applications for end users.

VI. IMPROVED COLLABORATIVE POLICY ADMINISTRATION

- ➢ ICPA Model: The proposed Improved Collaborative Policy Administration consists of two main stages, collaborative policy design and collaborative policy verification.

***Definition 1*:** Collaborative Policy Administration Model is,

$$ICPA :\{ Admins, CDM, CVM\}$$

Here, Admins refers to all involved policy administrators, like end users, developers, and marketers in the Applications.

Definition 2: Collaborative Policy Design Model is,

$$CDM :\{ PBhist, SimF, SUB, RefF, \Delta, P_{ref}\}$$

A policy administrator Admins can acquire a refined policy set ⊆ Pref according to a refinement function ⊆ RefF, which is a refinement driven by history data. In Definition 2, PBhist means to a policy base that contains a various policies previously created by administrator itself. PB_{hist} refers as,

$$PB_{hist} := 2_{SUB\_PER}$$

Here, SUB means to the subjects in a system. For example, all applications belong to SUB. PER means to all available permissions. Also SimF selects similar subjects, then produce their policies according to the subject's attributes as the similar policies.

Formally, $SimF: SUB \times PB_{hist} \rightarrow P_{similar}$

Here, P similar means to all various policies of the similar subjects. RefF means to the refinement functions, each one of which will output a policy set according to the attributes of a subject ∈ SUB, its similar policies ∈ Psimilar, and δ∈Δ, which may be a number.

$$RefF : SUB \times P_{similar} \times \Delta \rightarrow P_{ref}$$

Here, Pref is ⊆ SUB ×PER.

Definition 3: Collaborative policy verification model:

$$CVM :\{PBhist, SimF, SUB, VeriF, VeriR\}$$

A policy administrator ∈ Admins can obtain a verification result ∈ VeriR for a target policy set ∈ Ptarget, which contains all polices assigned to a target subject ∈ SUB, according to a verification function ∈ VeriF.

Here, SUB means to the target subjects that will be verified. VeriF means to the verification functions, each one of which will verify the target policy set, move towards a verification result.

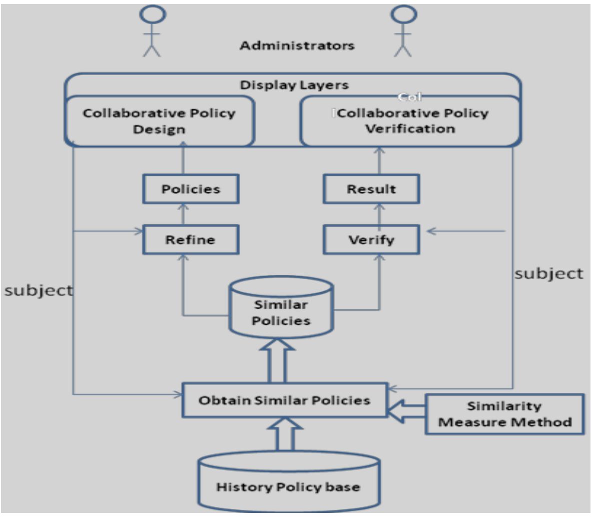
$$VeriF: VeriF: SUB \times P_{similar} \rightarrow VeriR$$

- Enforcement framework: In Fig 6 a policy administrator can leverage the framework to administrate policies via a web browser phone or development tool. The direction for key data flows is nothing but direction of arrows. Similarity measure methods and the history policy base are two key components in the enforcement framework. To impose CPA, the administrator should arrange a sufficient number of policies at first. Collaborative policy design, collaborative policy verification are the two key functions provided by the framework. These two functions depend on the history policy base as well as similarity measure methods. Then obtaining the similar policies, the two functions call a refinement algorithm, a verification algorithm. Finally, collaborative policy design and collaborative policy verification will display the output to the administrator on various user interfaces like development tool, a phone, web browser.
- Key Algorithms : To impose ICPA, similar policies algorithms, refinement algorithm, and verification algorithm are proposed as follows:

  - ✓ *Similar Policy Algorithm:*

Each similar policies algorithm obtains a similar policy set according to an input subject. If for every policy in the HB, every similar policies algorithm decides whether its subject is similar to the required subject, then add it to the similar policy set.

Fig 6 : Enforcement framework of ICPA.



A novel text mining technique to obtain similar policy sets of applications in Algorithm 1. This novel technique leverages the explanation of a target application to search similar applications, and then adds the requested permissions of the similar applications to the similar policy set of the target application. A TF-IDF method is engaged to create key words of application description, and then scores will be produced according to the key words. Finally, the novel technique chooses a predefined number (threshold) of applications according to the scores. At the end adds the chosen application policy configurations to the similar policy set.

Algorithm 1. Obtain Policies Based on Text Mining Method

```
Input:
        subject ∈ SUB
        HB ∈ $PB_{hist}$
Output:
        simpolicies ∈ $P_{similar}$
         initialize ( )
        query ← parse (subject.description)
        for all subject ∈ HB do
        doc ← subject.description
        score ← $a \times b \times \sum_{term \in query} (c \times d \times e \times f)(doc,term)$
        if score > simSubjs [simcountThreshold].score then
        simSubjs.removeLast() simSubjs.insertIndescendingOrderByScore (subject)
         end if
        end for
        for all subject ∈  simSubjs do
        simpolicies.add (subject.permissions)
        end for
        return simpolicies
```

In this algorithm, the initialize function engage declaring the simpolicies, assigning 0 to the score of each element in simpolicies and building the index for all application description from HB if the index files are not available. The parse function tokenizes the explanation of the subject and returns a query object that is ready for searching. The statements inside the for loop are created by a typical text mining procedure based on TF-IDF.

- ✓ Refinement Algorithm**:**

Algorithm 2 gives refinement policies according to a parameter δ, where δ is a number The time complexity is O (n), where, n means the number of policies in similar policies.

Algorithm 2. Collaborative Policy Refinement.
Input:

```
subject ∈ SUB
simpolicies ∈ $P_{similar}$
 $\delta \in \Delta$, it is a number
```

Output:

```
refpolicies ∈ $P_{ref}$
for all policy ∈ simpolicies do
 count [policy. permission] ++
end for
for all permission $\delta \in$ PERM do
 if count [permission]/simpolicies.size > $\delta$
policy.subject ← subject
 policy.permission ← permission
 refpolicies.add (policy)
end if
 end for
```

- ✓ Verification Algorithm:

Algorithm 3 gives a quantified measure between the target policies and similar policies. Time complexity of this algorithm is O (n), where n means to the size of similar policies, because the size of similar policies is normally larger than the size of target policies. Also the step to fetch target policies can be optimized by using an index of subjects in HB. The final result is a vector of percentages, that means much percentage the permission of target policy take up in the similar policies. To get simplified final result, design an aggregation algorithm to achieve a single number rather than a vector.

Algorithm 3: Collaborative Policy Verification
Input:

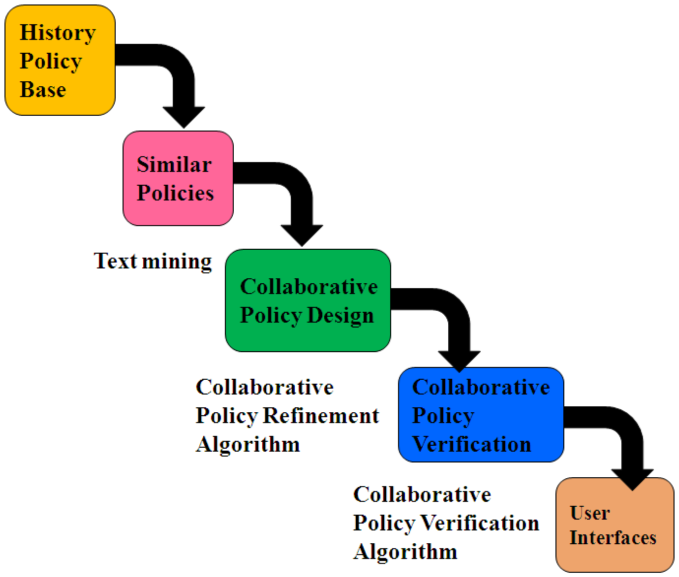
```
subject ∈ SUB
simpolicies ∈ $P_{similar}$
```

Output:

```
verifies ∈ VeriR
for all policy ∈ simpolicies do
 count [policy.permission] ++
end for
 targetpolicies ← ∀p ∈ HB: p.subject =
for all tpolicies ∈ targetpolicies do
 verires [tpolicy.permission] ←
 count [tpolicy.permission]/simpolicy
end for
```

VII. EXPERIMENTAL WORK

In ICPA to simplify the policy administration can refer to other similar policies to set up their own policies to protect privacy and other sensitive information. o obtain similar policies more effectively, a text mining-algorithm will be used. Then for enhancing design of policies refinement algorithm will be used. At the end to confirm the result verification algorithm will be used. Finally, collaborative policy design and collaborative policy verification will display the output to the administrator on various user interfaces like development tool, a phone, web browser.


History Policy Base
Similar Policies
Text mining
Collaborative Policy Design
Collaborative Policy Refinement Algorithm
Collaborative Policy Verification
Collaborative Policy Verification Algorithm
User Interfaces

VIII. CONCLUSION

The work presented in this paper proposes a novel policy administration mechanism, ICPA, to meet the requirements of the changing trust model, which has led to the widespread overclaim of privileges. ICPA leverages the similar policies to design or verify a target policy set, and simplifies the policy administration. This work provides definition of the formal model of ICPA and also the design of enforcement framework. Additionally, proposes text mining-based method of similarity measure to obtain similar policies. For future scope Safety definition is investigated and evaluated to improve permission model. For analysis of ICPA, more strengthening is required for mathematics depth.

ACKNOWLEDGEMENT

It is great pleasure for our, to acknowledge the assistance and contribution of number of individuals who helped us in presenting "Privacy monitoring in the online system using collaborative strategy management" . I take this occasion to thank God, Almighty for blessing me with his grace and taking our endeavor to a successful culmination. I extend our sincere thanks to our respected Head of the Department and our guide. We wholehearted thank to our family members, who are all support to do this work efficiently.

References

[1] IEEE transactions on parallel and distributed systems, 2013 Collaborative Policy Administration" by Weili Han, Member, IEEE, Zheran Fang, Laurence Tianruo Yang, Member, IEEE.
[2] "A Policy Management System for Collaborative Applications "by Mouhsine Lakhdissi, Hamid Harroud, AhmedKarmouch, Cliff Grossner, IEEE 2001.
[3] 2008 International Conference on Information Security and Assurance "Collaborative Privacy Management System" by In Joo Jang, Wenbo Shi, Hyeong Seon Yoo.
[4] International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), Orlando, Florida, USA,
[5] A survey on policy languages in network and security by Weili Han Chang Leianagement.
[6] IEEE on knowledge and data engineering, vol. 25, no. 7, july 2013" Multiparty Access Control for Online Social Networks" by Hongxin Hu, Member, IEEE, Gail-Joon Ahn, Senior Member, IEEE, and Jan Jorgensen
[7] A.K. Bandara, N. Damianou, E.C. Lupu, M. Sloman, and N. Dulay, "Policy Based Management," Handbook of Network and System Administration, Elsevier , Nov. 2007.
[8] D. Verma, "Simplifying Network Administration Using Policy- Based Management," IEEE Network, vol. 16, no. 2, pp. 20-26, Mar./ Apr. 2002.
[9] R. Yavatkar, D. Pendarakis, and R. Guerin, "A Framework for Policy-Based Admission Control," RFC 2753, no. 2753, 2000.
[10] B. Moore, E. Ellesson, J. Strassner, and A. Westerinen, "Policy Core Information Model—Version 1 Specification," IETF, RFC 3060, http://www.ietf.org/rfc/rfc3060, Feb. 2001.
[11] W. Enck, M. Ongtang, and P. McDaniel, "Understanding Android Security," IEEE Security & Privacy, vol. 7, no. 1, pp. 50-57, Jan./ Feb. 2009. [12] B. Sarma, N. Li, C. Gates, R. Potharaju, C. Nita-Rotaru, and I. Molloy, "Android Permissions: A Perspective Combining Risks and Benefits," Proc. 17th ACM Symp. Access Control Models and Technologies, pp. 13-22, 2012