King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

A new privacy preserving technique for cloud service user endorsement using multi-agents

D. Chandramohan [*], T. Vengattaraman, D. Rajaguru, P. Dhavachelvan

Dept of Computer Science, Pondicherry University, Pondicherry, India

Received 22 March 2013; revised 7 April 2014; accepted 4 June 2014
Available online 17 November 2015

KEYWORDS

Privacy preserving;
Petri-net;
Cohesion;
Authorization;
Multi-agents;
Web service

Abstract In data analysis the present focus on storage services are leveraged to attain its crucial part while user data get compromised. In the recent years service user's valuable information has been utilized by unauthorized users and service providers. This paper examines the privacy awareness and importance of user's secrecy preserving in the current cloud computing era. Gradually the information kept under the cloud environment gets increased due to its elasticity and availability. However, highly sensitive information is in a serious attack from various sources. Once private information gets misused, the probability of privacy breaching increases which thereby reduces user's trust on cloud providers. In the modern internet world, information management and maintenance is one among the most decisive tasks. Information stored in the cloud by the finance, healthcare, government sectors, etc. makes it all the more challenging since such tasks are to be handled globally. The present scenario therefore demands a new Petri-net Privacy Preserving Framework (PPPF) for safeguarding user's privacy and, providing consistent and breach-less services from the cloud. This paper illustrates the design of PPPF and mitigates the cloud provider's trust among users. The proposed technique conveys and collaborates with Privacy Preserving Cohesion Technique (PPCT), to develop validate, promote, adapt and also increase the need for data privacy. Moreover, this paper focuses on clinching and verification of unknown user intervention into the confidential data present in storage area and ensuring the performance of the cloud services. It also acts as an information preserving guard for high secrecy data storage areas.

1. Introduction

Contemporary IT-research makes web users share their resources from anywhere and everywhere through service-computing using cloud technologies. The emerging and vast growing cutting edge information technologies is paving way toward the next level of computing by utilizing software, hardware, operating systems, and all expected IT services globally in a matter of time with an affordable cost, with the help of user convenient devices throughout the world connected using

[*] Corresponding author.
E-mail addresses: pdchandramohan@gmail.com (D. Chandramohan), vengattaraman.t@gmail.com (T. Vengattaraman), raja.guru42@gmail.com (D. Rajaguru), dhavachelvan@gmail.com (P. Dhavachelvan).

Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2014.06.018

cloud computing. User behavior regulation has been chosen as one main strategic element by the content protection technologies. Privacy became one of the values embedded in content protection system design. In addition to the development of the content protection, technology can respond to privacy protection requirement in a goal oriented approach. Privacy no longer means anonymity/secrecy, when it comes to safeguarding of people's private communication and financial information (Facebook Vows to Fix Major Privacy Breach, 2011).

Next generation privacy preserving models and its principles have been already implemented by a few organizations for the sake of economic cooperation and development, especially, Asia pacific economic cooperation, United States federal trade commission, European Union Privacy directive, and federal and state/provincial laws in many countries. Self regulatory regions and industry serve as the starting point of protection around the world. But the realities of data fueled economy require a re-examination of how to implement a principle in a way that almost effectively serves the consumers. Privacy policy for private and government sectors are set on to implement a technologically advanced framework to protect highly confidential information stored in the cloud (Google to pay, 2012).

Some research labs framed its objectives to add value for framing a generic framework for blocking the breach happening and the privacy preservation development is to achieve and to protect privacy in significant ways, their objectives are to optimize the use of data for the benefit of both individuals and society, ensure that those data are accountable for its use, provide a regime that permits more effective oversight by regulators, and work effectively in a modern connected society. A data rich world requires numerous user controls and transparency features for both cloud users and providers to achieve privacy preserving objectives.

The end-user's valuable data are processed and stored in the cloud with different geographical locations. The leading service provider gives access to the storage as-a-services through their software as-a-services. User's information is under serious issue by unauthorized accesses. It is vulnerable if the secret data get compromised. Moreover, third party service providers are fond of user's private information for their business. It took place in a few computations, despite retrieving data from the storage services. It is a prime factor for every Cloud Service Provider to ensure the confidentiality of the user's private and personal data. To preserve data, the provider adopts their own framework and maintains the privacy of the registered users.

This paper is organized as follows. Section 2 presents the study of similar work and study identified in the cloud to preserve the privacy of the user data. Section 3 describes the formulation of Petri-net Privacy Preserving Framework and its layers. Section 4 presents the workflow of the framework and focus on Synchronization, Sequentiality, Concurrency and Conflicts (2S2C) approach. It is also to focus on the framework feasibility and its efficiency in the cloud environment. Section 5 presents an evaluation of experimental results analysis and its comparison. Finally, Section 6 presents the conclusion part with future key factor to carry the research further.

2. Related and background work

In cloud data storage, privacy preserving is one apex concern in today's emerging IT world. Many researchers have been targeting this field. Liu et al. (2012), investigate the characteristics of cloud storage services and propose a secure and privacy preserving keyword searching scheme. This allows the Cloud Service Providers (CSP) to participate in the decipherment, and to return only files containing certain keywords specified by the users. His team focused on reducing both the computational and communication overhead in decryption for the user's data, on the condition of preserving user data privacy and user querying privacy. Hao et al. (2011), propose a remote data integrity checking protocol that supports data dynamics.

It supports public verifiability. The proposed protocol supports public verifiability without the help of a third-party auditor. Wang et al. (2011), studied the problem of ensuring the integrity of data storage in Cloud Computing. The task of allowing a third party auditor, on behalf of the cloud client, is to verify the integrity of the dynamic data stored in the cloud. The authors found it is critical to enable a Third Party Auditing to evaluate the service quality from an objective and independent perspective. Zhang et al. (2012), check the customer's need to take certain actions to protect their privacy with noise injection. Service providers will be confused about which requests are real ones. The authors develop a novel historical probability based noise generation strategy. It generates noise requests based on their historical occurrence probability so that all requests including noise and real ones can reach about the same occurrence probability, and then service providers would not be able to distinguish them. Wang et al. (2011), proposed an approach to solve the problems of privacy and security by including access control for the encrypted data, and revoking the access rights from users when they are no longer authorized to access the encrypted data.

The authors propose a hierarchical attribute based encryption scheme, by combining a hierarchical identity based encryption system and a cipher text-policy attribute-based encryption system. Liu et al. (2009), investigated the characteristics of cloud computing and proposes an efficient privacy preserving keyword search scheme in cloud computing and it enables the service provider to search the keywords on encrypted files to protect the user data privacy and the user queries privacy efficiently. Public Key Encryption and decryption techniques adapted in this paper provide privacy in cloud. It allows the service provider to participate in partial decipherment to reduce a client's computational overhead. It is semantically secure. Itani et al. (2009), present Privacy as a Service (PasS) a set of security protocols for ensuring the privacy and legal compliance of customer data in cloud computing architectures. PasS allows for the secure storage and processing of users' confidential data by leveraging the tamper-proof capabilities of cryptographic coprocessors. The author uses tamper-proof facilities to provide a secure execution domain in computing cloud that is physically and logically protected from unauthorized access. Author achieved user-configurable software protection and data privacy mechanisms by his proposed approach.

Wang et al. (2010), explain that in each cloud service it will exchange data with other clouds, so when the data are exchanged between the clouds there exists the problem of disclosure of privacy. Privacy disclosure problem about individual or company is inevitably exposed while releasing or sharing data in the cloud service. This paper suggests some privacy preserving technologies used in cloud computing services. The author argued that it is very important to take privacy into account when designing cloud services. Zhou et al. (2010), found that the concerns are not adequate and more should be added in terms of five aspects (i.e., availability, confidentiality, data integrity, control, and audit) for security.

Released acts on privacy to protect users' private information in the new environment are out of date. The author studied adapting released acts for new scenarios in the cloud, which will result in more users to step into cloud. Pearson (2009), discusses the privacy challenges that software engineers face when targeting the cloud as their production environment to offer services are assessed, and key design principles are suggested. The author explains the risks to privacy mitigated and that data are not excessive, inaccurate or out of date, or used in unacceptable or unexpected ways beyond the control of data subjects. Many authors propose a privacy approach to prevent users' valuable information in cloud data center (Chandramohan et al., 2012a,b; 2013).

Huang et al. (2010) found an interactive protocol and an extirpation based key derivation algorithm combined with lay revocation, multi-tree structure and symmetric encryption to form a privacy preserving, effective framework for cloud storage area. Dhasarathan et al. (2014) Prefaces a validating policy to safeguard the user data by a mathematical distributed approach for breach less cloud service in all circumstances without effecting the service providers efficiency. Li et al. (2011) Global Enforcement of Data Assurance Control (GEODAC) framework is proposed to assure data enforcement globally by a policy approach. It preserves the data retention, data migration, and data appropriateness which are stored in cloud. Moreover, the policy is represented by a state of lifecycle stages and a state machine based representation. Wang et al. (2014) secure watermark detection is described in a compressive sensing based framework using multiparty computation protocol (MCP) under semi-honest security model to preserve the confidential information in the cloud storage region.

To hide private data from the unauthorized services and users, an interactive protocol is designed to resolve the cloud storage privacy preservation. A key derivation algorithm is adopted to generate and manage keys of the data owners and storage service providers (Huang et al., 2011). The data ownership to avoid the anonymous authentication based on public key cryptography, and a tunable k-control trade-off between the degree of anonymity and the computational overhead were imposed by the system. In which, it would be a control system framework for the cloud users (Khan and Hamlen, 2012). To personalize the computing by intelligent processing in hybrid cloud, by predicting the user activity and their interventions are monitored using the privacy framework (Zhang et al., 2013). A virtual application with customized security policies are adopted to provide such services in a preventable approach (Zhao et al., 2012).

To maintain the user secrecy and leverage the need of confidentiality prevention a complete study has been deliberated which proposed a framework to prevent the information (Wei et al., 2012). To reduce the data redundancy and data duplication in the cloud efficient block encryption and duplication algorithms are used to design a privacy preserving framework in Nimgaonkar et al. (2012). Moreover, to reduce the computation complexity a key proxy re-encryption is used. CTrust framework for ubiquitous access restriction used a secure hyper visor as a building block to prevent the storage area discussed in Lin et al. (2013). This framework is working with partial trust on service providers. A proxy based framework was proposed by a team of researchers in Singhal et al. (2013) for preserving mobile health monitoring system by coupling with decryption technique. In the cloud, services are leveraged as storage, network and servers, which are provided by platform as a service.

Ray and Biswas (2014) described the cryptographic solution for preserving the security of healthcare service customers by HIPAA policy. Moreover, Al-Muhtadia et al. (2011) maintain a threshold limit for a ubiquitous environment using cryptography techniques. Debnath et al. (2014) show the advantage of ring signature as a digital verification to prevent the unknown user's intrusion in the sensor networks.

3. Proposed approach: PPPF

The development of computing technology evolved through cloud computing. The whole IT world, academic sector, finance sector, government sector, and health care system have adapted cloud services in their work area. Users may access their data anywhere when in need of it. Cloud computing delivers their request as in the form of services to them. One can keep away from owning huge storage area and maintaining it by storing their data in the cloud. User's data privacy became a question mark by deploying their personal database in the cloud. Data stored in the service provider's end is highly risky because anyone can identify and collect one's personal information and it may lead to privacy bleach. Unknown cloud users may cause leakage in personal information by regular monitoring and collecting data regarding the client. The proposed framework acts as a secrecy locker for cloud users and providers. The PPPF focuses on layered approach which incorporates the traditional state transition representation with compressive data handling, mutual service oriented structure, unauthorized user detection key encryption handling and decryption identification to preserve users confidentiality in the cloud environment.

Privacy strategists are dealing today with multitudinous devices, applications and networks that need to be secured. There may be several ways to secure applications. These include web application firewalls, real time application monitoring and two factor authentication. Cloud organizations must secure users data at the application, endpoint, storage area and device levels, etc., the providers need to find the right balance between privacy and flexibility. Some service provider environments may rob an organization's data with their flexibility. Since there are too many devices to control, securing access has become a top priority for cloud providers for organizations.

Maintaining secrecy of user's information is one of the major issues in cloud computing. The secrecy of user database should be maintained properly, or else information gets breached consistently. We came forward with a privacy preserving framework to solve privacy issues. Layered privacy approach may be a way to detect and isolate unusual threats. We are focusing on an integrated layered set up for proposing the privacy preserving framework. It is essential to protect privacy of one's information in the cloud data storage. A few notable areas where the privacy breach happens are,

- API-interfaced application infection (Third Party Interfering)
- Privacy data loss in mass storage area (Distributed Server Storages)
- Privacy at service provider level (Policy Framing and Organizing)

- Privacy at users/client level (Responsibility and supporting providers by reducing unknown identifications)
- If the providers are not accomplishing any one of generic universal standard and unique service level agreement for service providers and harsh cipher laws like European Union, United State and Switzerland (*EU-US-SWIS*) on intruders it might procure to data lose EU-US-SWIS {*European Union, United State and Switzerland*}

In such a Framework setup, each layer overlaps the previous layer. In this manner, whatever gets missed in the first layer is caught by the second. To describe secrecy protocols for each and every application in the cloud may bring in a certain amount of rigidity into the process of delivering IT-services. So the purpose of this paper is to propose a model for user entry-level restrictions for cloud service using Petri-net distribution model, and a set of privacy metrics for proposed user entry-level restriction. Finally this paper concludes by suggesting a privacy enforcement exploiter authentication technique for the cloud.

4. Proposed PPPF workflow model

Organization's personal data get unruffled and upheld communally, and used by providers without the knowledge of the cloud users. It is a violation of confidentiality within users and may lead to a huge exposure of private data in the IT world. These users trust their service providers and share their precious information. It has been noticed from the literature study of privacy preserving techniques in cloud data storage, that it adopted some hand full of privacy policy to protect the user's data from breaching. Those policies are claimed to be more rigid because of the policy framing strategies. It was adopted from The United State and The European Union (US-EU) privacy policy.

Researchers ardently elicit the origin for privacy breaching happening in and around the IT world since even the leading cloud providers failed to accept their user's privacy kept confidentially (Google, Amazon, SalesForce.com, VMware, Dropbox, Social Networking providers, etc.) (Facebook Vows to Fix Major Privacy Breach, 2011; Google to pay, 2012; LinkedIn Corp, 2012; Dropbox User, 2012). It is analyzed and targeted to light up the user's privilege to possess and furnish to set their privacy and endorsement of priceless data. PERMIS authentication technique (Chadwick and Fatema, 2012) gave a vague idea for researchers to concentrate on this big issue. It is presumed to have an influence on the whole IT industry, E-governance, government secret information, business, healthcare, individual privacy right, etc., as a landmark to impede these issues and prevent all secret data leverage and its breaching out. In this paper we are going to propose a generic privacy preserving authentication approach shown in Fig. 1 with cohesive Petri-net modeling and we designed a framework using it to develop this loom. Our framework consists of seven different modules inbuilt with four cohesive Petri-net modules to surmount a silhouette.

In this section, we discuss the Petri-net Privacy Preserving Framework, the main components of this system include Cloud Service Request, User Validation, User Request Verification, Cloud requestors Authorization and Cloud users Authentication or response, which are presented in Fig. 2 and described in detail below (Chandramohan et al., 2012a,b; 2013).

- Cloud Service Provider
- Cloud Service Request
- Petri-Net Privacy Preserving Model
- User Request Verification
- User Validation
- Cloud requestors Authorization and
- Cloud users Authentication or response

Preserving one's data in cloud before getting invaded was a risky responsibility for both providers and users. Fig. 3 gives an invasion mitigation technique to minimize the risk factor and develop a rigid trust on cloud providers.

```
Method Type $CP_r\_CR_q()$
BEGIN
   Get i/p for r and q
   $CP_r: = $ Manipulate $(CP_s)$
   $CP_s$ checked with $CR_q$ and verify for Trust Policy $T_{pi}$;
For $(CP \neq 0)$
   Do until $(\{CR_q == Tpi\ [CPi]\})$
   Return value for CP $(T_{pi})$:
If $(CP \rightarrow (CR_q <\ > 0))$ then
   State 1 = $P_i$;
   $CR_q$ should satisfy Petri-net policy $P_i$
End
If $(CRq = True)$ then verified and filter to next validation
   State 2 = $Pi * (R_q * V_n * V_t * CP_{pi}))$;
   Repeat until delivers TRUE;
End
If $(CR_q = NP)$ then {New Policy (NP)}
   State 3 = $TN_{pi}$;
End
If $(ST_n ==$ Return 1) then {State (ST)
   IFF $(\{ST_n = (ST_1 * ST_2 * ST_3)\})$ {IFF-if and only if}
   $ST_n$ = Always Return 1;
End
Else
If $(TR_{pi} = E_x S_{pi})$
   Then Validate & Authenticate:
   $SR_i \rightarrow \{TR_{pi}, P_{si}, E_x, S_{pi}\}$;
   Entrée to Data;
   $R_q \rightarrow$ Recognized as authenticated user;
End
Else
   $R_q$ = RETURN 0;
   Exit No Authorization;
   End If
   End For
End
```

A generic flow carried out in PPM Cloud Provider CP_r and CR_q begin the process with input request ‘q’ and response with ‘r’. It manipulates $CP_r: = CP_s$ and similarly it checked with ‘CR_q’ and verify for Trust Policy ‘Tp_i’ is available or not. If it is pre-defined with policy then it get verified until ‘CP’ refined to null until ‘CR_q’ should satisfy Petri-net policy ‘Pi_+’. It is prolongs the same stratagem until it complete the execution ‘CR_q’ turned to be true. Rapidly verify and filter to next validation State $T_2 = Pi * (Rq * Vn * Vt * CPpi)$, once these steps get athwart then repeat until it delivers TRUE values. Now we arrived to end the initial state. To carry forward the initial true values there presents few pre-conditions as quantitative measures CRq = NP and it mitigates the privacy policy with

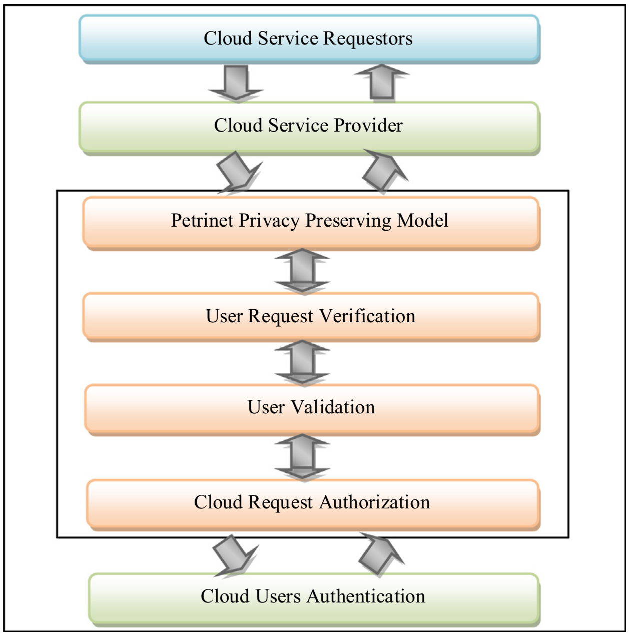

Cloud Service Requestors
Cloud Service Provider
Petrinet Privacy Preserving Model
User Request Verification
User Validation
Cloud Request Authorization
Cloud Users Authentication

Figure 1 Petrinet privacy preserving framework-PPPF.

EU-SW laws. A New Policy (NP) is framed from the initial rule CRq then TN_{pi} policy is derived after nominal standard are inherited from the available measures. Once policies get validated continue the verification whether ST_n returns to be true.

If and only if $ST_n = ST_1 * ST_2 * ST_3$, STn will be always true, otherwise $TR_{pi} = E_xS_{pi}$ is Validated & Authentication process starts from SRi → $\{TR_{pi}, P_{si}, E_x, S_{pi}\}$. After all these truncation processes if any request gets passed by returning a true value, 'R_q' he/she can be allowed or Recognized as authenticated user to view the stored data and information (Chandramohan et al., 2012a,b; 2013). If any one of the above processes failed and is noticed to obtain a false state immediately the whole system gets truncated and 'Rq' response is 0 i.e. if no Authorization, he/she is rejected for requested service and sent out from regular cycle. Simultaneously a log file is maintained to verify and identifying if any user repeating the vulnerable activities in future. If they are found to be one among them they are punished according to the EU-policy and law. The representation of PPPF-Petri-net Privacy Preserving Framework is designed and structured to handle the complex interaction with cloud requester and provider. Our proposed framework has the ability to identify the Synchronization, Sequentiality, Concurrency and Conflicts of different cloud users to access their own data without disturbing other cloud users' information. So many new researches have been progressing to preserve the privacy of cloud user's information. We came up with PPPF framework as a milestone to achieve preserving user information in a cloud environment. Each provider has their own privacy policy and law to protect their data storage area located worldwide. One can climb that the existing policies are not adequate to preserve users' confidentiality from the recent incident noticed from world fames service provider's like Facebook, Google, LinkedIn and Dropbox etc., (Facebook Vows to Fix Major Privacy Breach, 2011; Google to pay, 2012; LinkedIn Corp, 2012; Dropbox User, 2012) the existing policies are not adequate to preserve users' confidentiality. The proposed mitigation flow persuades to prevent user's information from an unknown users grab.

Cloud Request Providers CP_r riposte as per the customary granted Trust Policy T_{pi} while $CP \neq 0$ && $CP = Ø$. The retort capitulation repeats until $CR_q < > 0$. CRq tartan its medley with Petri-net policy 'P_i' and stick to set its conduit by (R_q, V_n, V_f and CP_{pi}). To isolate consequently with conciliation rule framed from TN_{pi} and to aim at the data availability for stipulate users without compromising personal data. If and only if all former steps get processed, it gets verified correctly and the output response returns to be true. Otherwise it truncates farther processing into the data storage area.

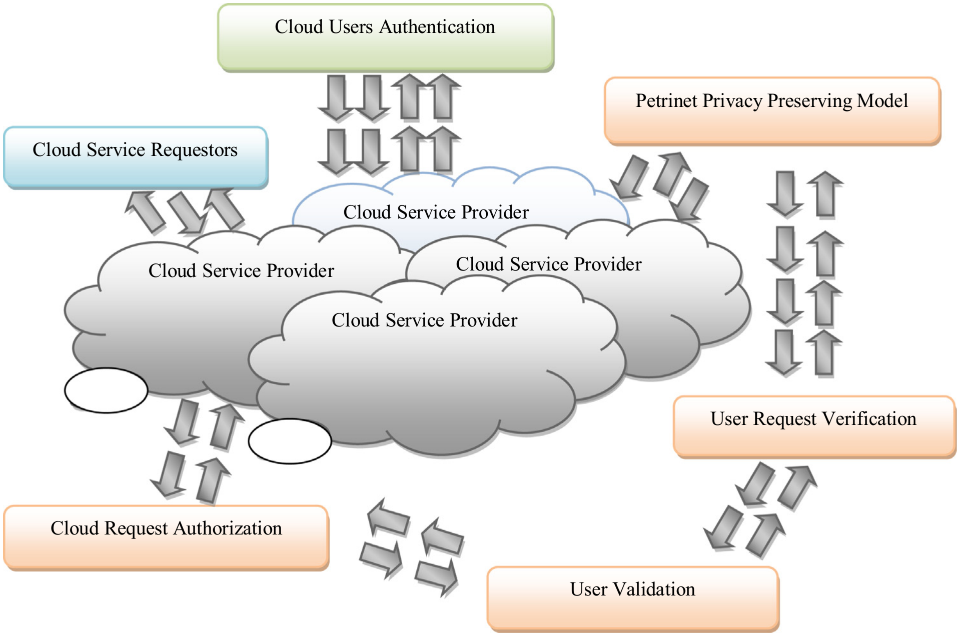


Figure 2 Petrinet privacy preserving work flow in cloud.

Similarly two factor authentication starts with TR_i and E_xS_{pi} to refine the consumer policy according to the proposed method. It permits requested users by limiting their accessibility and secrecy priority. It reiterates the whole method for accessing cloud data storage area and allocates confidential information to the correct user (Chandramohan et al., 2012a, b; 2013).

Figs. 3 and 4 shows current progression works under the umbrella of algorithm stated below and these steps are followed continuously until the user gets identified. Their original data are kept more confidential. Start with Request to the Cloud Service Providers as $\{CSR_i\}$Send Request to $\{CSP_i\}$, repeat request until it returns Concrete solution. $\{RR_i\}$ Repeat $\neq Ø$ {Until} $RR_i \,\Omega\, E_s(CSP_i - CSR_i)$ {E_s-EXPECTED SERVICE}, $E_s(CSP_i - CSR_i)$ Expected service gets salvage prop up then, go to the previous and promote state of affairs to obtain the truthful user. $CSP_i \geqslant \sum \{CU_{ai} + PPM_i\,(UR_v + U_v + CR_{ai})\}$. This shows the Privacy Preserving Algorithmic approach for the proposed framework in Figs. 1 and 4 and its internal doling out with the help of Fig. 5. Leading research scholars deal with this issue to enhance the privacy features as a deterrent footstep for the preservation of user's data .

Cloud users' verification and their validation carried out with PPPF and its seven different modules are designed as high cohesion intra-modules that shall have an influence on the next module authentication and carry forward the request query. Clients need authorization from cloud providers to get their quantifiable services. $\{CP_{ai}\}$ Repeating a verification course of action in all intra-modules with cohesion principle, $NA_i \pm Q_s\yen\,(CSP_i + CSR_i)$ {Q_s-Quantifiable Service}. These evaluation factors are manipulated by $Q_s\,(CSP_i + CSR_i)$, For $NAi \pm Qs\yen\,(CSP_i + CSR_i)$ where $i = \{0, 1, 2, 3 \ldots n\}$; *Iff* it-may get assorted as per cloud user notations, constraint and attributes exploits by particular authentication evolution waiting to reach the final destination CSR Ω CSP && $CSR_i <\ > CSP_i$.

```
Step 1: $CRP_n = UR_i + \sum(\{Rsz_n * Rsq_n * Rcy_n * Rcf_n\})$
Step 2: if 2S2C $Rsz_n$ = TRUE; then goto Step 6:
RETRUN 1; Else
Step 3: $CRP_n = UR_i + \sum(\{Ø * Rsq_n * Rcy_n * Rcf_n\})$
Step 4: $CRP_n = UR_i + \sum(\{Ø\})$
Step 5: $CRP_n = \{Ø\}$; Return 0; End; // $\{0, 1, 1, 1\}$; or $\{0, 0, 0, 0\}$;
or $\{0, \infty, \infty, \infty\}$;
Step 6: $CRP_n = UR_i + \sum(\{Rsz_n \neq Ø * Rsq_n * Rcy_n * Rcf_n\})$
Step 7: $CRP_n = UR_i + \sum(\{1 * Rsq_n * Rcy_n * Rcf_n\})$
Step 8: if 2S2C step 7 $\neq Ø$; continue with residual 7 PPCT
modules to Return:1;
//$\{1, 1, 1, 1\}$; or $\{1, 0, 0, 0\}$; or $\{1, \infty, \infty, \infty\}$;
Else goto step: 5. Return: 0; End;
```

The following Petri-net Preserving Framework properties have to solve the complexity among the interactions through 4 basic self requirements and Fig. 3 all the providers' and clients' request should be communicated through these principles namely synchronization (Rsz), sequentiality (Rsq), concurrency (Rcy) and conflicts (Rcf).

- Liveness – Cl
- Safeness – Cs
- Boundedness – Cb
- Conservation – Cv
- Reachability – Crc
- Place Invariant – Cpivt
- Priority Levels – Cpl
- Reliability – Crty

```
Step 1: Start with Request to the Cloud
   Service providers as
   $\{CSR_i\}$Send Request to $\{CSP_i\}$
Step 2: Repeat request until it returns
   Concrete solution.
   $\{RR_i\}$ Repeat $\neq Ø$ {Until}
   $RR_i \,\Omega\, E_s\,(CSP_i - CSR_i)$
   {$E_s$-EXPECTED SERVICE}
Step 3: $E_s\,(CSP_i - CSR_i)$ Expected service
   get salvage prop up then,
   go to step 5:
Step 4: Starts a loop for
   $CSP_i \geqslant \sum \{CU_{ai} + PPM_i(UR_v + U_v + CR_{ai})\}$
   do ...
   The process as per user authentication,
   End; Return 1: Repeat step 3 until CSPi reached an
authorization state
Else
   Exit; Return 0: the Request process through valid
exception and followed below, do
Step 5:CSRΩCSP end exit
$CU_{ai} \rightarrow$ Cloud User Authentication,
$UR_v \rightarrow$ Requested User Verification,
$U_v \rightarrow$ User Validation,
$RR_i \rightarrow$ Repeat Request
$CR_{ai} \rightarrow$ Cloud Requestor authorization,
PPMi $\rightarrow$ Layers framed using $\{UR_v, U_v, \text{and } CR_{ai}\}$
```

```
Step 1: Begin Cloud Service
   providers service to users requests
   $\{CSP_i\}$Send Response to $\{CSR_i\}$
   $CSR_i <\ > CSP_i$
Step 2: Clients need authorization from
   Cloud providers to get their quantifiable services.
   $\{CP_{ai}\}$ Repeat 1: do
   $NA_i \pm Q_s\yen(CSP_i + CSR_i)$
   {$Q_s$-Quantifiable Service}
Step 3: $Q_s(CSP_i + CSR_i)$;
Step 4: Iff only then
   $CSP_i \geqslant \sum \{NA_i + MA_i(UR_v + U_v + CR_{ai})\}$
   Continue step2 and end it
   else
   For $NA_i \pm Q_s\yen(CSP_i + CSR_i)$
   $i = \{0, 1, 2, 3, \ldots n\}$;

   i- may get varied as per their notations and parameters and attributes.
Step 5: Authorization get verified as per their validation measures, verification
techniques, quality measure, scaling measure, etc.

Step 6:End; Return $\leqslant 1$: Repeat step 2 until
   CSRi gained an authentication state
   Else end Return null
   Exit; the Request process by through valid exception and
Step 7:CSRΩCSP&& $CSR_i <\ > CSP_i$
End; $NA_i \rightarrow$ Need Authorization, $MA_v \rightarrow$ Maturity Verification,
```

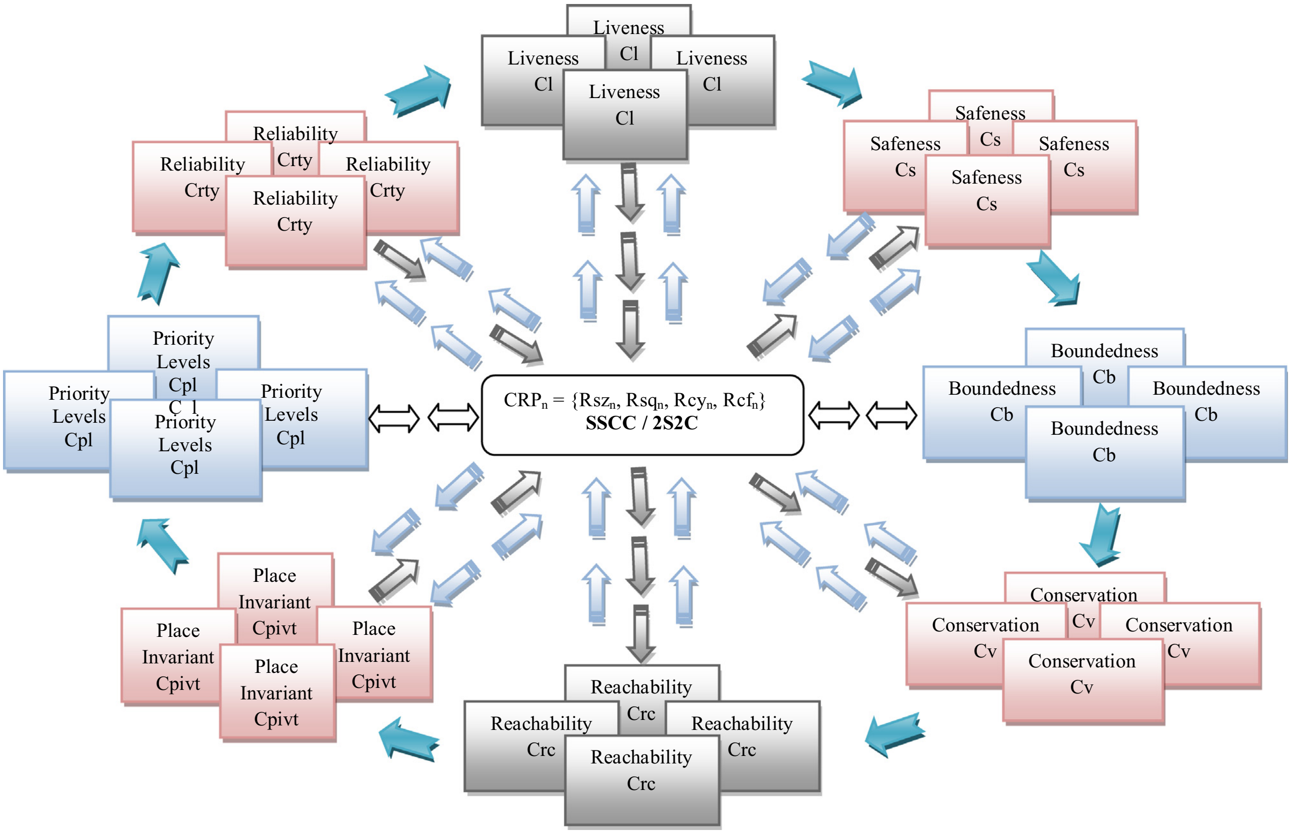


Figure 3 2S2C privacy preserving cohesion technique.

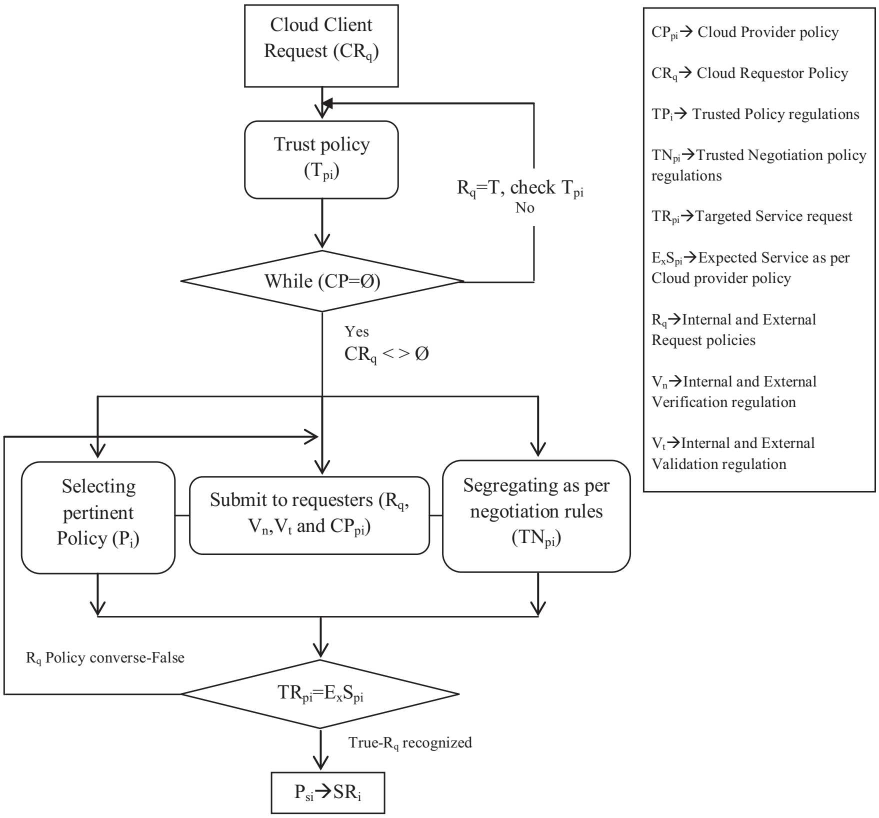

Cloud Client Request (CR_q)
Trust policy (T_{pi})
R_q=T, check T_{pi}
No
While (CP=Ø)
Yes
$CR_q <> Ø$
Selecting pertinent Policy (P_i)
Submit to requesters (R_q, V_n, V_t and CP_{pi})
Segregating as per negotiation rules (TN_{pi})
R_q Policy converse-False
$TR_{pi}=E_xS_{pi}$
True-R_q recognized
P_{si}→SR_i
CP_{pi}→ Cloud Provider policy
CR_q→ Cloud Requestor Policy
TP_i→ Trusted Policy regulations
TN_{pi}→Trusted Negotiation policy regulations
TR_{pi}→Targeted Service request
E_xS_{pi}→Expected Service as per Cloud provider policy
R_q→Internal and External Request policies
V_n→Internal and External Verification regulation
V_t→Internal and External Validation regulation

Figure 4 Sequential dynamic privacy preserving cloud service flow.

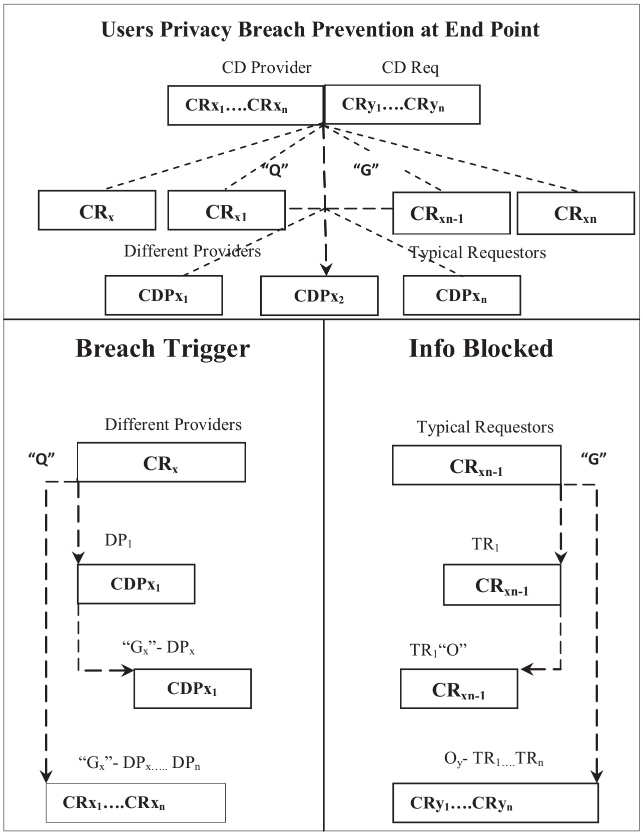


Figure 5 Privacy breach prevention at cloud user's end point.

Preliminaries

Cloud Request Provider ($\mathrm{CRP_n}$) = {$\mathrm{Rsz_n}$, $\mathrm{Rsq_n}$, $\mathrm{Rcy_n}$, $\mathrm{Rcf_n}$}

Synchronization (Rsz), Sequentiality (Rsq), Concurrency (Rcy) and Conflicts (Rcf) (2S2C), where the request gets varied accordingly form 0 to n and it is denoted as Request synchronization various form (0–n), Request Sequentiality various form (0–n), Request Concurrency various form (0–n) and Request Concurrency various form (0–n), where the 2S2C delivers its cohesive nature form this scenario by comparing with each service request and the condition applied by the provider. In this section we introduce the origin of 2S2C into the proposed framework as an ordinal highly cohesive module to verify user identification and their originality. Moreover, the dynamic event driven function with the priorities and decision making are carried out by definition the policy with set of rules and axioms as a prevention measure.

Axiom 1

$$\text{Cloud Request Provider }(\mathrm{CRP_n}) = \{\mathrm{Rsz_n}, \mathrm{Rsq_n}, \mathrm{Rcy_n}, \mathrm{Rcf_n}\} \tag{1}$$

Definition 1. Requester's inputs are mounted to verify and validate the user's identity for accessing information in the cloud using our proposed **SSCC** pre-requesting privacy cohesion technique. SSCC (Synchronization, Sequentiality, Concurrency and Conflicts also named as **2S2C** technique)

$$\left.\begin{array}{ll}\mathrm{Rcy_n(Rsq_1)} = \{\mathrm{Rsz_1}\} & \mathrm{Rcf_n(Rsq_1)} = \{\mathrm{Rsz_2}, \mathrm{Rsz_3}\} \\ \mathrm{Rcy_n(Rsq_2)} = \{\mathrm{Rsz_2}\} & \mathrm{Rcf_n(Rsq_2)} = \{\mathrm{Rsz_4}\} \\ \mathrm{Rcy_n(Rsq_3)} = \{\mathrm{Rsz_3}\} & \mathrm{Rcf_n(Rsq_3)} = \{\mathrm{Rsz_5}\} \\ \mathrm{Rcy_n(Rsq_4)} = \{\mathrm{Rsz_4}\} & \mathrm{Rcf_n(Rsq_4)} = \{\mathrm{Rsz_2}\} \\ \mathrm{Rcy_n(Rsq_5)} = \{\mathrm{Rsz_4}, \mathrm{Rsz_5}\} & \mathrm{Rcf_n(Rsq_5)} = \{\mathrm{Rsz_1}\}\end{array}\right\} \tag{3}$$

Definition 2. *Sequential Execution*: The sequential execution (SE) and its execution $\mathrm{S_j}$ can fire only after the firing of Si. This imposes the precedence constraints $\mathrm{S_i}$ & $\mathrm{S_j}$. Such precedence constraints are typically of the execution as a part of a dynamic system.

$SE \overset{N-1}{\rightarrow} \{S_i, S_j\}$ *Iff* $\mathrm{S_j}$ starts functioning only when $\mathrm{S_i}$ is done with its verification.

Definition 3. *Synchronization System*: In the proposed system policy integration and verification of highly cohesive instances are synchronized for execution of predefined process to covenant with multiple real-time systems. It set the state '$\mathrm{S_z}$' to get enable only when two different executions are triggered simultaneously and request '$\mathrm{S_t}$' to set all (z′, z″) for all expected possible results. $S_z \overset{z',z''}{\Longleftrightarrow} \{S_t\}$.

Definition 4. *Concurrency Identification*: $\mathrm{C_y}$ deposits and verifies user request in two or more places to deliver correct user interaction $\mathrm{t_i}$, $\mathrm{t_j}$ and System interaction ($\mathrm{SI_t}$) $\mathrm{S_i}$, $\mathrm{S_j}$.

$$C_y \rightarrow (SI_t((S_i, S_j) * (t_i, t_j))) \tag{4}$$

Definition 5. *User Conflict*: If the user's response probability distributions noticed to get conflict with actual state ($\mathrm{Cl_i/Cl_j}$) thereafter the request might turn off the initial demand and continue with the reachable operation.

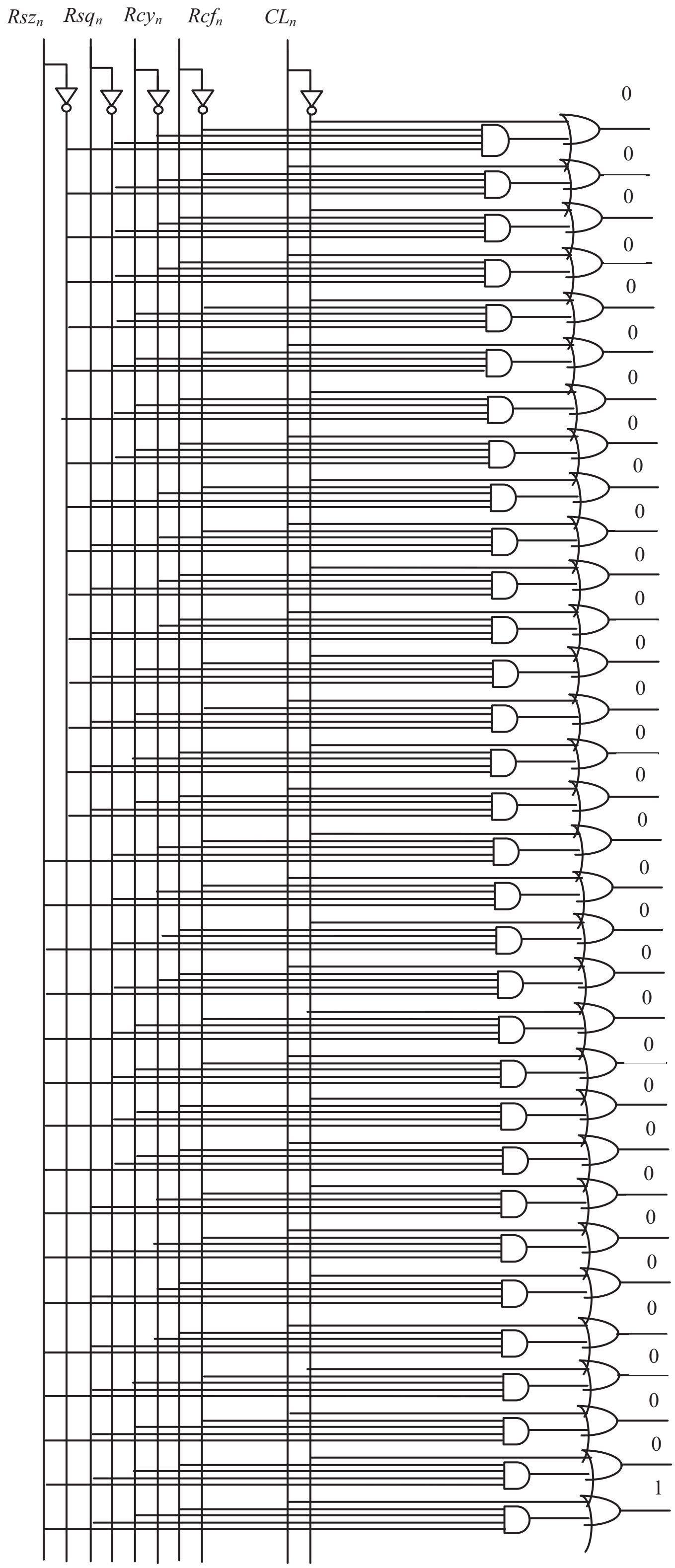
```
If ($l_i$ = +ve) then

Continue to next position verification state;
Deactivate "$l_j$", {$l_i$ = Ø}; Return Ø;
Else
($l_i$ = +ve) make $l_i$ as + ve State;
{$l_j$ = Ø}; Deactivate $l_i$; & Return Ø: procedure;

End If
```

$\mathrm{C_f} = +\mathrm{ve} : (\mathrm{Cl_i} : \mathrm{Cl_j});$

$$\left.\begin{array}{l}\mathrm{Rsz_n} = \{\mathrm{Rsz_1}, \mathrm{Rsz_2}, \mathrm{Rsz_3}, \mathrm{Rsz_4}, \mathrm{Rsz_5}\}; \\ \mathrm{Rsq_n} = \{\mathrm{Rsq_1}, \mathrm{Rsq_2}, \mathrm{Rsq_3}, \mathrm{Rsq_4}, \mathrm{Rsq_5}\}; \\ \mathrm{Rcy_n} = \{\mathrm{Rcy_1}, \mathrm{Rcy_2}, \mathrm{Rcy_3}, \mathrm{Rcy_4}, \mathrm{Rcy_5}\}; \\ \mathrm{Rcf_n} = \{\mathrm{Rcf_1}, \mathrm{Rcf_2}, \mathrm{Rcf_3}, \mathrm{Rcf_4}, \mathrm{Rcf_5}\};\end{array}\right\} \textbf{2S2C} \text{ Privacy Preserving Cohesion Technique} \tag{2}$$



Step 1: CLn input is checked with 2S2C technique if it returns 1 go to next level else

Step: 2 If the resultants of step 1 return 0: then CLn check with next set of combinations until it returns 1 to make its true combination with next set of inputs.

Step 3: Repeat step 1 and 2: for all set of Modules in 2S2C technique and make sure until it returns 1.

Step 4: As per the 2S2C-PPPF algorithm, the input condition check with all combinational logic and returns false (0) and one combination return true (1)

Figure 6 2S2C privacy preserving logic diagram.

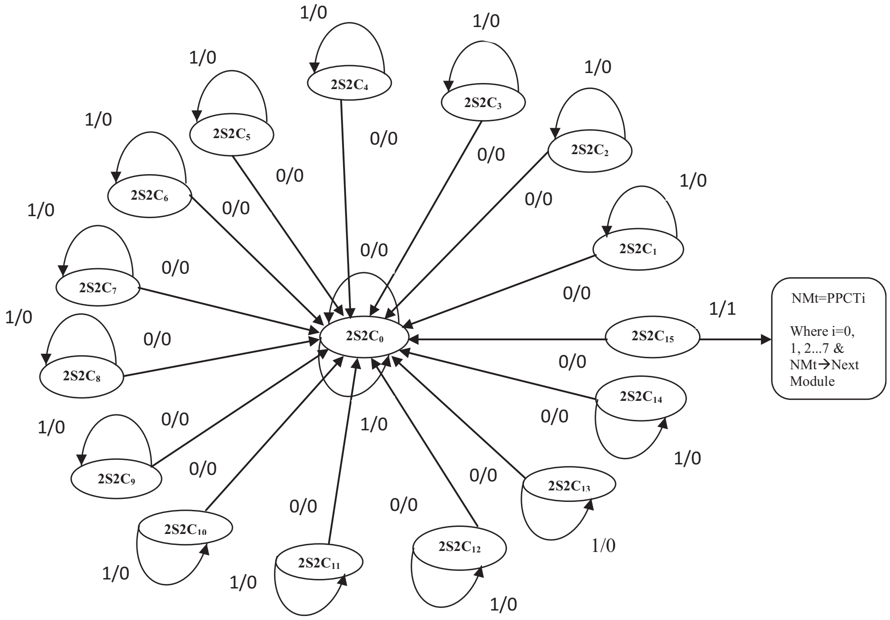
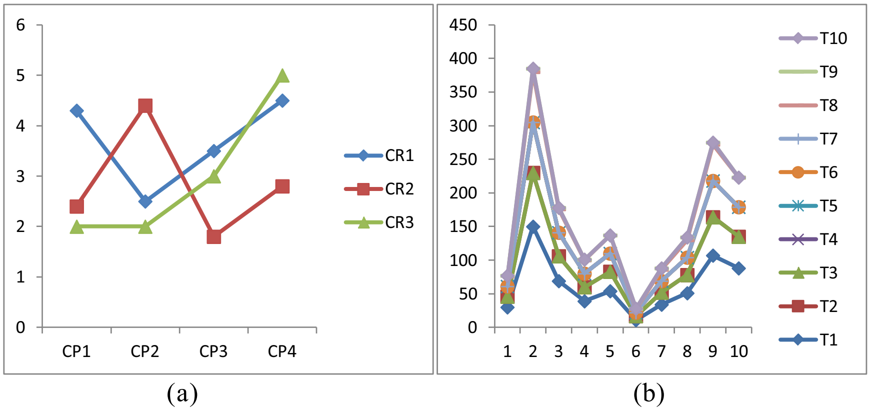


Figure 7 State transition diagram representation of 2S2Ci-modules in PPCT.



(a) (b)

Figure 8 (a, b) PPPF verification and validation of minimal-support T-Invariants and Linear Combinational construction of privacy conflict in cloud data storage.

In order to define the axioms of 2S2C its attributes are debut by {SE, Sz, Cy, Cf} respectively, (Synchronization (SE), Sequentiality (S_z), Concurrency (C_y) and Conflicts (C_f)). The privacy preserving module is defined as PPM_i validated with respect to i value which can be varied from i = (1 to n). In 2S2C scenario n has the maximum probability of 1 to 8. Similarly the liveness of first privacy module verification of user request and their data is represented as (PC_{L0}). Their internal function f is calculated as per the transactions 't' and the number of places 'p' required to complete a task f (T_n, P_n).

Axiom 2

We define 2S2C as an axiom in our proposed PPPF using definitions 1, 2, 3, 4, and 5. It should act consequently with different request and repeat the execution and generate its outcome represented in axiom 2.

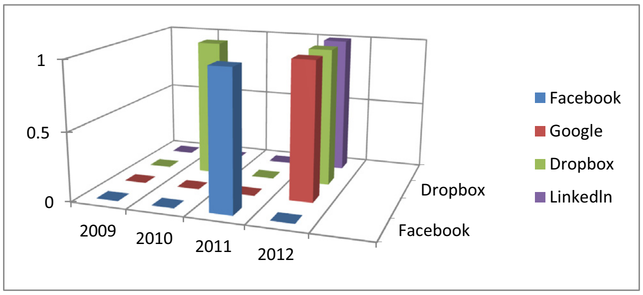


Figure 9 Privacy breach identified in recent years arise from leading Cloud Service Providers.

Resolution for Conflicting Transitions	Probability	Priority
T1	0.3	0.8
T2	0.5	0.6
T3	0.5	0.9
T4	0.5	0.1
T5	0.6	1.6
T6	0.8	2.9
T7	0.1	3.7
T8	0.5	4.2
T9	0.7	9.0
T10	0.3	12

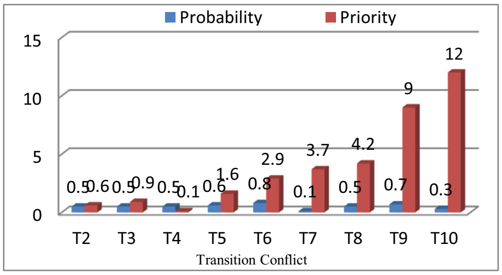


Figure 10 PPPF probability and priority privacy transition conflict in cloud data storage.

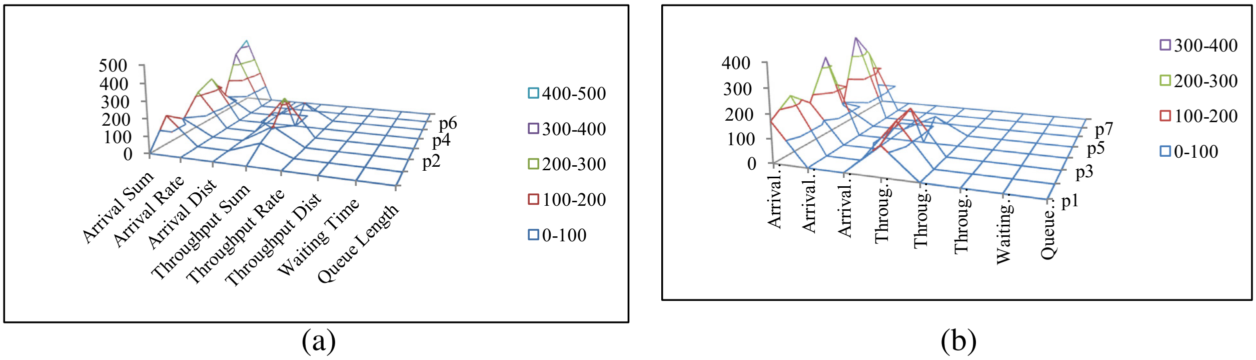


Figure 11 (a,b) Data privacy breach Identification and its effective prevention at storage area using PPM.

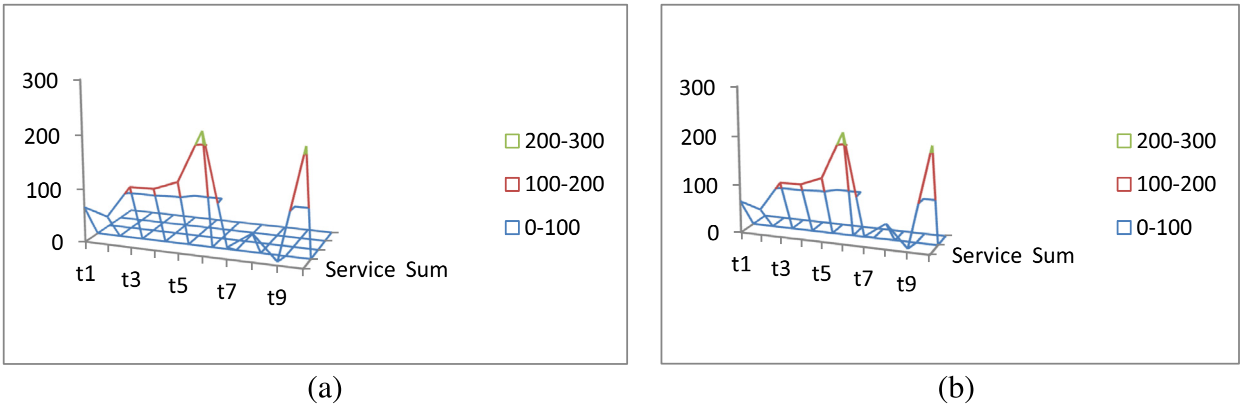


Figure 12 (a, b) PPPF users secrecy prevention during data transitions request and response in the cloud.

$$\{\mathrm{SE},\mathrm{S}_z,\mathrm{C}_y,\mathrm{C}_f\}\rightarrow\left\{\begin{array}{l}\mathrm{Rsz}_n=\{Rsz_1,Rsz_2,Rsz_3,Rsz_4,\mathrm{Rsz}_5\};\\ \mathrm{Rsq}_\mathrm{n}=\{\mathrm{Rsq}_1,\mathrm{Rsq}_2,\mathrm{Rsq}_3,\mathrm{Rsq}_4,\mathrm{Rsq}_5\};\\ \mathrm{Rcy}_n=\{\mathrm{Rcy}_1,\mathrm{Rcy}_2,\mathrm{Rcy}_3,\mathrm{Rcy}_4,\mathrm{Rcy}_5\};\\ \mathrm{Rcf}_\mathrm{n}=\{\mathrm{Rcf}_1,\mathrm{Rcf}_2,\mathrm{Rcf}_3,\mathrm{Rcf}_4,\mathrm{Rcf}_5\};\\ SE\overset{N-1}{\rightarrow}\{S_i,S_j\};\\ S_z\overset{z',z''}{\Longleftrightarrow}\{S_t\};\\ C_y\rightarrow\left(SI_t\left((S_i,S_j)*(t_i,t_j)\right)\right);\\ \mathrm{C}_{\mathrm{f(i,j)}}=+\mathrm{ve},\ [\mathrm{Cl}_i=\varnothing],\ [\mathrm{Cl}_\mathrm{j}=\varnothing];\end{array}\right\} \tag{5}$$

By converting into mathematical form 'f', $\because$ $\mathrm{f(2S2C)}\rightarrow\mathrm{f}(\mathrm{T_n},\mathrm{P_n})=2^\mathrm{n}$

$\mathrm{f}(\mathrm{T_n},\mathrm{P_n})=2^\mathrm{n}$ where $\mathrm{n}=\{0,1,2,\text{ and }3\}$ respectively as per 2S2C.

$$\{\mathrm{f}(\mathrm{T_n},\mathrm{P_n})=2^\mathrm{n}\}\Rightarrow\{\mathrm{f}(\mathrm{T_0},\mathrm{P_0})=2^0,\ \mathrm{f}(\mathrm{T_1},\mathrm{P_1})=2^1,\ \mathrm{f}(\mathrm{T_2},\mathrm{P_2})=2^2,\ \mathrm{f}(\mathrm{T_3},\mathrm{P_3})=2^3\}$$

$$\{\mathrm{f}(\mathrm{T_n},\mathrm{P_n})=2^\mathrm{n}\}\Rightarrow\{1,2,4,8\}$$

Theorem 1. Let PPPF as $\mathrm{PPM_i}$ protect the sensitive data present at the data center, by blocking multiple unknown users' hands on confidential information stored in the cloud. $\mathrm{PPM_i}$ framework modules are communal to verify and are checked with privacy **2S2C** technique at each and every level.

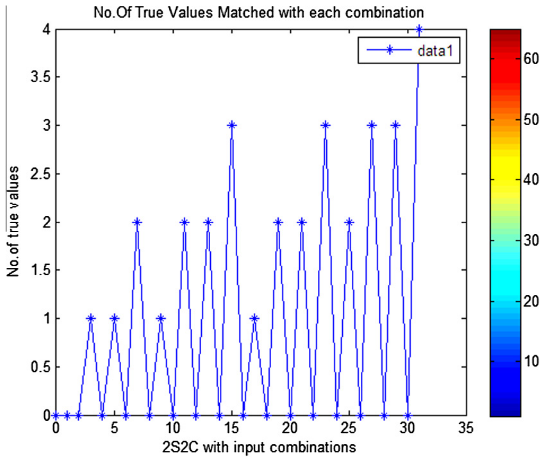


Figure 13 2S2C combinations in logical truth value portrayal.

User identification management gets interacted and authorized according to the policy agreement between cloud users and providers. The bonding among sub-system structure modules are intra-dependent on each other so each input and output sub tasks are dependent on 2S2C evaluation.

Proof. Now, PPMi modules are verified with 2S2C cohesion technique in all possible conditions (initial, typical, custom, and medium). Let's check the trial and error method to verify the possibility of getting penetrated or blocking user into source data 'D_i'.

PPM_i here let's consider $i = N$, where $\{N = (n + 1)\}$,

$PPM_i = P_n + f(T_n, P_n)$.

Liveness- C_l

In 2S2C $\{Rsz_n, Rsq_n, Rcy_n, Rcf_n\} = 2^n$ where $n = \{0, 1, 2, 3 \ldots, n\}$ technique users input (information) parameters are checked periodically with Cloud Request Provider (CRP_n) validating their suitability of accessing information stored in the cloud. The user's request UR_i communicates with

Table 2 Recent confrontation identified in leading cloud service providers.

Recent confrontation	Facebook	Google	Dropbox	Linkedln
2009	0	0	0	0
2010	0	0	1	0
2011	1	0	0	0
2012	0	1	1	1

Table 1 2S2C privacy preserving verification and validation.

Present state				Input	Next state				Output
Rsz_n	Rsq_n	Rcy_n	Rcf_n	C_{Ln}	$Rsz_n * C_{Ln}$	$Rsq_n * C_{Ln}$	$Rcy_n * C_{Ln}$	$Rcf_n * C_{Ln}$	$CRP_n = C_{Ln} + \sum(\{Rsz_n * Rsq_n * Rcy_n * Rcf_n\})$
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	1	1	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0
0	0	1	1	1	0	0	1	1	0
0	1	0	0	0	0	0	0	0	0
0	1	0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0	0	0
0	1	0	1	1	0	1	0	1	0
0	1	1	0	0	0	0	0	0	0
0	1	1	0	1	0	1	1	0	0
0	1	1	1	0	0	0	0	0	0
0	1	1	1	1	0	1	1	1	0
1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0
1	0	0	1	0	0	0	0	0	0
1	0	0	1	1	1	0	0	1	0
1	0	1	0	0	0	0	0	0	0
1	0	1	0	1	1	0	1	0	0
1	0	1	1	0	0	0	0	0	0
1	0	1	1	1	1	0	1	1	0
1	1	0	0	0	0	0	0	0	0
1	1	0	0	1	1	1	0	0	0
1	1	0	1	0	0	0	0	0	0
1	1	0	1	1	1	1	0	1	0
1	1	1	0	0	0	0	0	0	0
1	1	1	0	1	1	1	1	0	0
1	1	1	1	0	0	0	0	0	0
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

To reach the confidential data area Table 1 illustrates the accessibility of a user to storage area would be verified with present data request, Inputted data and final reachable stage of a user. It is identified based on the defined privacy preserving policy adopted using 2S2C by verifying the identity of user at each and every stage of the authentication process.

Table 3 PPPF-minimal-support T-Invariants and Linear Combinational construction verification of privacy conflict in cloud data storage.

Minimal support T-Invariants	Linear Combinations constructed									
T1	30	150	69	39	54	11	34	51	107	88
T2	16	80	37	21	29	6	18	27	57	47
T3	0	0	0	0	0	0	0	0	0	0
T4	15	75	35	20	27	5	17	26	54	44
T5	0	0	0	0	0	0	0	0	0	0
T6	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	0	0	0	0
T8	15	75	35	20	27	5	17	26	54	44
T9	1	5	2	1	0	1	2	4	3	0
T10	0	0	0	0	0	0	0	0	0	0

Table 4 PPPF resolution for conflicting transitions, probability and priority of privacy conflict in cloud data storage.

Resolution for conflicting transitions	Probability	Priority
T1	0.3	0.8
T2	0.5	0.6
T3	0.5	0.9
T4	0.5	0.1
T5	0.6	1.6
T6	0.8	2.9
T7	0.1	3.7
T8	0.5	4.2
T9	0.7	9.0
T10	0.3	12

the cloud through our proposed technique as a privacy preserving measure, according to our approach the request is sent to PPCT-Privacy Preserving Cohesion Technique that consist of eight different modules PPM_i where i = {0, 1, 2, …, 7}, each and every input module gets hold of validation with the liveness module of 2S2C technique C_l, later its output acts as an input to subsequent modules. If liveness is inequitable, it throws its first exception info and exits from its farther symmetric cycle.

Let us consider the single sequence input request carried out inside 2S2C with zeroth module 'C_l'.

$$\begin{aligned} &PPM_i = PC_{L0} \\ &(PC_L)_{n+1} = PC_{Ln} + f(T_n, P_n) \\ &(PC_L)_{n+1} = MP_n + f(T_n, P_n) \because \mathit{MP_n - Current\ Module\ of\ PPM\ i.e.\ PC_{Ln}} \end{aligned} \tag{6}$$

Initially start with n = 0,

$PC_{L0+1} = MP_0 + f(T_0, P_0) \because f(T_n, P_n) = 2^n$ *according to Axiom* 2

$PC_{L1} = MP_1 + 2^0$

$PC_{L1} = 1 + 1$

$PC_{L1} = 2;$

The result shows positive implication with 2S2C single input, similarly we have to verify with the rest three inputs,

Now put n = 1 in Eq. (6),

$PC_{L1+1} = 2 + f(T_1, P_1)\ [\because PC_{L1} = 2]$

Table 5a Service Provider Privacy Breach limitations identified at storage area using PPM.

Place name	Arrival sum	Arrival rate	Arrival dist	Throughput sum	Throughput rate	Throughput dist	Waiting time	Queue length
p1	171	0	0	125	0	0	0	0
p2	237	0	0	176	0	0	0	0
p3	171	0	0	188	0	0	0	0
p4	341	0	0	0	0	0	0	0
p5	88	0	0	75	0	0	0	0
p6	385	0	0	0	0	0	0	0
p7	280	0	0	0	0	0	0	0

Table 5b Global Privacy Breach Prevention in data storage area using PPM.

Place name	Arrival sum	Arrival rate	Arrival dist	Throughput sum	Throughput rate	Throughput dist	Waiting time	Queue length
p1	167	0	0	66	0	0	0	0
p2	95	0	0	107	0	0	0	0
p3	219	0	0	229	0	0	0	0
p4	266	0	0	0	0	0	0	0
p5	114	0	0	105	0	0	0	0
p6	368	0	0	0	0	0	0	0
p7	441	0	0	0	0	0	0	0

$\mathrm{PC}_{\mathrm{L}2} = 2 + 2^1$ [$\because f(T_n, P_n) = 2^n$ *according to Axiom* 2]
$\mathrm{PC}_{\mathrm{L}2} = 4$;Now put n = 2 in Eq. (6),
$\mathrm{PC}_{\mathrm{L}2+1} = 4 + \mathrm{f}\ (\mathrm{T}_2, \mathrm{P}_2)$ [$\because \mathrm{PC}_{\mathrm{L}2} = 4$]
$\mathrm{PC}_{\mathrm{L}3} = 4 + 2^2$ [$\because f(T_n, P_n) = 2^n$ *according to Axiom* 2]
$\mathrm{PC}_{\mathrm{L}3} = 8$;Now put n = 3 in Eq. (6),

$$\begin{aligned}
&\mathrm{PC}_{\mathrm{L}3+1} = 8 + \mathrm{f}(\mathrm{T}_2, \mathrm{P}_2)[\because PC_{L3} = 8] \\
&\mathrm{PC}_{\mathrm{L}4} = 8 + 2^3[\because \mathrm{f}(\mathrm{T}_\mathrm{n}, \mathrm{P}_\mathrm{n}) = 2^\mathrm{n}\ \mathit{accordingtoAxiom2}] \\
&\mathrm{PC}_{\mathrm{L}4} = 16; \\
&\{\mathrm{PC}_{\mathrm{L}1}, \mathrm{PC}_{\mathrm{L}2}, \mathrm{PC}_{\mathrm{L}3}, \mathrm{PC}_{\mathrm{L}4}\} = \{2, 4, 8, 16\}
\end{aligned} \tag{7}$$

$$C_{L0} = \left\{ \begin{matrix} \begin{bmatrix} 0 & 1 & 11 \\ 0 & 0 & 00 \\ 0 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 11 \\ 0 & 0 & 00 \\ 0 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 01 \\ 0 & 0 & 00 \\ 0 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 00 \\ 0 & 0 & 00 \\ 0 & \infty & \infty\infty \end{bmatrix} \end{matrix} \right. \qquad C_{L1} = \left\{ \begin{matrix} \begin{bmatrix} 0 & 1 & 11 \\ 1 & 0 & 00 \\ 0 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 11 \\ 1 & 1 & 00 \\ 0 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 11 \\ 1 & 1 & 10 \\ 0 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 11 \\ 1 & 1 & 11 \\ 0 & \infty & \infty\infty \end{bmatrix} \end{matrix} \right.$$

$$C_{L3} = \left\{ \begin{matrix} \begin{bmatrix} 0 & 1 & 11 \\ 0 & 0 & 00 \\ 1 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 11 \\ 1 & 0 & 10 \\ 1 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 11 \\ 1 & 0 & 01 \\ 1 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 11 \\ 0 & 1 & 00 \\ 1 & \infty & \infty\infty \end{bmatrix} \end{matrix} \right. \qquad C_{L4} = \left\{ \begin{matrix} \begin{bmatrix} 0 & 1 & 11 \\ 0 & 0 & 10 \\ 1 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 11 \\ 0 & 0 & 01 \\ 1 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 11 \\ 0 & 1 & 11 \\ 1 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 11 \\ 0 & 1 & 10 \\ 1 & \infty & \infty\infty \end{bmatrix} \end{matrix} \right.$$

$$C_{L2} = \left\{ \begin{matrix} \begin{bmatrix} 0 & 1 & 11 \\ 0 & 0 & 00 \\ 1 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 11 \\ 1 & 0 & 00 \\ 1 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 11 \\ 1 & 1 & 00 \\ 1 & \infty & \infty\infty \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 11 \\ 1 & 1 & 10 \\ 1 & \infty & \infty\infty \end{bmatrix} \end{matrix} \right. \qquad C_{L5} = \left\{ \begin{bmatrix} 0 & 1 & 11 \\ 1 & 1 & 11 \\ 1 & \infty & \infty\infty \end{bmatrix} \right.$$

$\mathrm{CL}_\mathrm{n} \rightarrow$ Cloud Liveness
Where $\mathrm{n} = \{0, 1, 2, \ldots \mathrm{n}\}$

In $\mathrm{PC}_{\mathrm{Ln}}$, {2, 4, 8, 16} ⇒ $2\mathrm{S}2\mathrm{C}_\mathrm{i}$ = {($2\mathrm{S}2\mathrm{C}_0$, $2\mathrm{S}2\mathrm{C}_1$, $2\mathrm{S}2\mathrm{C}_2$, $2\mathrm{S}2\mathrm{C}_3$)}; where i = {0, 1, 2, 3}.

If and only if all the 2S2C inputs get verified and the results indicate a positive signal, then the 'PC_L' overall output is carried forward as an input value to PPM_i the subsequent next module. Similarly same process is repeated until PPM_i and 2S2C cohesive technique gets verified and indicates a positive response to the requested user. Immediately the user request gets quit from PPM_i workflow if any one of its module indicates a negative sign (i.e.) the request is identified to be unknown.

The 2S2C iteration and its module are verified with the rest of PPM_i form 1 to 8, Cloud request Safeness (C_s), Cloud Request Boundedness (C_b), Cloud Conservation (C_v), Cloud Request Reachability (C_{rc}), Cloud Request Place Invariant ($\mathrm{C}_{\mathrm{pivt}}$), Cloud Request Priority Levels (C_{pl}), Cloud Request Reliability ($\mathrm{C}_{\mathrm{rty}}$) and $2\mathrm{S}2\mathrm{C}_\mathrm{i}$.

Cloud providers ($\mathrm{CR}_{\mathrm{x}1} \ldots \mathrm{CR}_{\mathrm{xn}}$), Cloud requestors ($\mathrm{CR}_{\mathrm{y}1} \ldots \mathrm{CR}_{\mathrm{yn}}$), Different Providers are defined by CR_x $\mathrm{CR}_{\mathrm{x}1}$ and Typical Requestors (TR) are denoted as ($\mathrm{CR}_{\mathrm{xn}-1}$ $\mathrm{CR}_{\mathrm{xn}}$) and ($\mathrm{CR}_{\mathrm{xn}-1}$, TR_1, $\mathrm{CR}_{\mathrm{xn}-1}$, $\mathrm{CRx}_1 \ldots \mathrm{CRx}_\mathrm{n}$). TR_1"O" initialized with cloud requestors $\mathrm{CR}_{\mathrm{xn}-1}$, O_y-$\mathrm{TR}_1 \ldots \mathrm{TR}_\mathrm{n}$. with respect to "Q" CRx, DP1 and "G_x" DP_x $\mathrm{CRy}_1 \ldots \mathrm{CRy}_\mathrm{n}$.

5. Experimental methodology and result analysis

In this experimental methodology section, we first present the control logic flow representation for the 2S2C technique. It is then proceeded to necessary verification and validation for a trusted authorization, which supports a state transition modeling for the proposed cohesive technique. The experiment anal-

Table 6a Service Providers Privacy Breach limitations identified using PPPF.

Transition name	Service sum	Service rate	Service dist	Service time	Utilization
t1	66	0	0	0	0
t2	53	0	0	0	0
t3	113	0	0	0	0
t4	114	0	0	0	0
t5	131	0	0	0	0
t6	226	0	0	0	0
t7	22	0	0	0	0
t8	53	0	0	0	0
t9	7	0	0	0	0
t10	215	0	0	0	0

Table 6b PPPF Secrecy Prevention Verified at different State of Transition.

Transition name	Service sum	Service rate	Service dist	Service time	Utilization
t1	119	0	0	0	0
t2	84	0	0	0	0
t3	88	0	0	0	0
t4	88	0	0	0	0
t5	140	0	0	0	0
t6	148	0	0	0	0
t7	113	0	0	0	0
t8	83	0	0	0	0
t9	5	0	0	0	0
t10	132	0	0	0	0

Table 7 PPPF Coverability Tree – Text Mode M [p1, p2, p3, p4, p5, p6, p7]; M = [100, ω]; M0 = [100, 200].

From	Fired	To
M0	T0, T1	M1
M1	T1, T2	M2
M2	T2, T3	M3
M3	T3, T4	M4
...	...	...
...	...	...
...	...	...
Mn	T_N	M_{n+1}

Table 8 PPPF input invariance IM preservation time in mSec.

CA(n)	CP_n-Time Invariant in mSec
CA(0)	305.0035
CA(1)	205.0073
CA(2)	307.0037
CA(3)	127.0062
CA(4)	9.0037
...	...
...	...
CA_N	Cn1, Cn2, Cn3, ... Cn – 1, Cn, Cn + 1

ysis is followed with necessary comparison and evaluation parameters. The proposed flow is evaluated with the Petri-net (PN) tool to check its efficiency in normal, medium and critical scenarios. Then we compared the PPPF with the existing privacy frameworks and its features are illustrated with a table representation, which shows the PPPF implementation architecture performance is comparatively high in all scenarios (see Figs. 6–13).

Table 1 illustrates the clear mock-up identification of recent privacy breach happened globally by leading cloud providers. *Facebook* vows to fix major privacy breach-Australian report-sep-2011 (Facebook Vows to Fix Major Privacy Breach, 2011). *Google* pays $22.5 million to settle privacy charges: July-2012 *WSJ-Wall Street Journal* (Google to pay, 2012). *LinkedIn* sheds more light on Privacy Breach, san-fancisco: *LinkedIn* corps criticized for inadequate network security after hackers exposed millions of its user's passwords Jun-2012 (LinkedIn Corp, 2012). *Dropbox* confirms it got hacked, will offer two-factor authentication. Spammers used stolen password to access a list of Dropbox user e-mails. Aug-12 (Dropbox User, 2012). *Salesforce.com* sent an e-mail to its customers notifying them that a variety of recent phishing attacks against salesforce and officially confirmed they are hacked (www.zdnet.com) (see Table 2).

2S2 Ci → Where {i = 0, 1, 2...15}, PPCTi → Where {i = 0, 1, 2...7}, 2S2C – {Synchronization (Rsz), Sequentiality (Rsq), Concurrency (Rcy) and Conflicts (Rcf)}, PPCT-{Privacy Preserving Cohesion Technique}, NMt-Next Module present in PPCT-Technique.

All input requests are processed through the proposed system (2S2C-PPCT), where PPCT consists of eight different modules and 2S2C has four different qualitative attributes, these attributes are considered in digital logic combinational approach starting from 0 to 15, unerringly the system checks 16 different combinations i.e. (0000, 0001, 0010, ... 1101, 1110, 1111) and communicates with those four qualitative attributes. Two different input states {0, 1} are checked with these combinations to arrive at an authenticate and authorized state. It is explained in Fig. 5 with the help of a transition state diagram. Correspondingly Fig. 5 processes all eight different modules {Liveness-Cl, Safeness-Cs, Boundedness-Cb, Conservation-Cv, Reachability-Crc, Place Invariant-Cpivt, Priority Levels-Cpl, Reliability-Crty} in same way by communicating with {Rsz, Rsq, Rcy, Rcf}.

Table 10 Input invariance IM preservation time in mSec.

A(n)	Time invariant in mSec
A(0)	408.0059
A(1)	409.0059
A(2)	409.0063
A(3)	274.0084
A(4)	9.0054
...	...
...	...
A_N	n1, n2, n3, ... n – 1, n, n + 1

Table 9 Privacy preserving representation in complex logical interaction of PPPF.

UR	SB	SC	R	Consistent		Structural enabling bound (SE)									
				TI	PI	SE	SE	SE	SE	SE	SE	SE	SE	SE	SE
						T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
P1	B	X	**Y**	**Y**	**Y**	UD	UD	UD	UD	UD	UD	UD	UD	UD	UD
P2	**UB**	X	**Y**	**Y**	**Y**	**D**	**D**	UD	UD	UD	UD	**D**	UD	**D**	UD
P3	B	**Y**	**Y**	N	**Y**	UD	**D**	**D**	UD	**D**	UD	UD	**D**	UD	UD
P4	B	X	**Y**	N	**Y**	UD	UD	D	UD	UD	**D**	**D**	UD	UD	UD
P5	**UB**	**Y**	**Y**	N	**Y**	UD	UD	UD	D	UD	UD	UD	UD	**D**	UD
P6	**UB**	X	**Y**	N	**Y**	UD	UD	UD	UD	UD	**D**	**D**	UD	UD	UD
P7	**UB**	X	**Y**	N	**Y**	UD	UD	UD	UD	UD	**D**	**D**	UD	UD	**D**

5.1. Minimal-support T-Invariants

n-Rank (A) = 3 = > at most T-Invariants are linearly independent Linear Combinations constructed with these vectors are displayed after 2nd column (see Tables 3–9).

```
<?xml version="1.0"?>
<PNToolbox>
<PPMModel_name>PPM.xml</PPMModel_name>
<Type>2</Type> <!-- T-timed CPN-- >
<Seed>66</Seed> <!-- initialseed-- >
<Place> <!-- placedefinitionCRn -- >
<Id>p1</Id> <!-- place'sid CPn-- >
<Value>5,43</Value>
<Color>black</Color>
<Label>
<Name>Cp1</Name>
<Offset>0.50,-0.20</Offset>
<Visible>yes</Visible>
</Label>
<PPMInitialMarking>5</PPMInitialMarking>
<PPMCapacity>Inf</PPMCapacity>
</Place>
<PPMTransition>
<Id>Ct1</Id>
<Value>8,45</Value>
<Color>black</Color>
<PPMMessage>Firing transition Ct1</PPMMessage>
<Label>
<Name>t1</Name>
<Offset>0.41,-0.12</Offset>
<Visible>yes</Visible>
</Label>
<Time>
<PPMDistribution>constant</PPMDistribution>
<PPMParameters>3</PPMParameters>
</Time>
</PPMTransition>
<PPMTransition>
<Id>t2</Id>
<Value>6,37</Value>
<Color>black</Color>
<PPMMessage>Firing transition t2</PPMMessage>
<Label>
<Name>t2</Name>
<Offset>0.70,-0.34</Offset>
<Visible>yes</Visible>
</Label>
<Time>
<PPMDistribution>cont. uniform</PPMDistribution>
<PPMParameters>2.5,7</PPMParameters>
</Time>
</PPMTransition>
<PPMArc>
<Id>a1</Id>
<From>p1</From>
<To>t1</To>
<Style>1</Style>
<Type>1</Type>
<Color>black</Color>
<Weight>2</Weight>
</PPMArc>
<PPMArc>
<Id>a2</Id>
<From>p1</From>
<To>t2</To>
<Style>1</Style>
<Type>1</Type>
<Color>black</Color>
<Weight>3</Weight>
</PPMArc>
<PPMProbability>
<PPMTransitions>t1,t2</PPMTransitions>
<Values>0.25,0.75</Values>
</PPMProbability>
</PNToolbox>
```

User Request-UR; Bounded-B; UN-Bounded-UB; Structured Boundedness-SB; Structured Conservativeness-SC; True-Y; False-X; Repetitiveness-R; TI-T-Invariant; I-P-Invariant; Determined-D; Undetermined-UD; Consistent-C; Structural enabling Bound-SE.

An incidence Matrix form of cloud service exchange and user interaction is happening at different service request and its privacy verification and evaluation process is denoted in the form of a matrix $A_i\ (A_0 * A_i)$. Cloud service exchange is denoted by $P_n = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7\}$, Interaction service request and responses as $T_n = \{T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10}\}$ and its input variance $IM = \{A_i\ (A_0 * A_i)\}$

$$A_i\ (A_0 - A_i) \rightarrow IM = A_i\ (T_n * P_n)$$

$$IM = A_i\ (T_{10} * P_7)$$

Table 11 Different data privacy preserving frameworks in the cloud and their prime factors compared with PPPF.

Comparison of frameworks	GEODAC	DPPCSF	CS-MPCF	PPPF
Policy based approach	√	X	X	√
Symmetric key encryption	X	√	X	√
Key derivation algorithm	X	√	X	√
State machine representation	√	X	X	√
Petrinet layers	X	X	X	√
Watermark detection	X	X	√	X
Compressive sensing	X	X	√	X
Cohesive technique	X	X	X	√

$$A_i = (10 * 7) \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 15 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A_0 = (10 * 7) \rightarrow \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$A = (A_0 - A_i) \rightarrow \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -15 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Table 10 shows the timed invariants and its iterations happen in these stipulated time intervals from different users to Cloud Service Providers.

Designs of privacy preserving cloud storage framework, GEODAC framework and a compressive sensing based framework have been proposed literally to preserve user information in the cloud. Huang et al. (2010) concentrated on symmetric key encryption algorithm by clustering with lazy revocation, multi-tree structure and extirpation based key derivation algorithms for designing and developing an encryption based system. Li et al. (2011) framed a policy based privacy framework to preserve data in the cloud. Wang et al. (2014) studied the multimedia privacy issue and developed a compressive sensing based framework using MCP which protects semi-trust users. PPPF demonstrates the data privacy potency in a cloud storage area with Petri-net based cohesive framework to preserve and prevent the cloud user's data privacy. The paper analyses the effectiveness of PPPF and its feasibility by comparing existing frameworks in Table 11. PPPF identifies the un-trusted users and voids their services if they are trying to access the private information stored in the cloud.

6. Conclusions

This paper discusses the need for a generic privacy preserving framework, which performs a decisive task in preserving user's confidential data, which is stored in the cloud storage service provider. The Gargantuan rise in the cloud service era, may lead to users losing control over the storage environment. However, to satisfy the ever-growing concerns of user's requirement and the expected services and their valuable data pertained system utilization explores to limitless service (Multi-specialty software's, Applications, Platforms, Entertainment, E-governance and so on). Cloud users are compelled to share their complete niceties and information to the providers by accepting cloud provider's terms and conditions. Only 5% to 10% of the users are aware of the fact that the provider has access to their personal information. This is a serious issue in the emerging cloud storage world. This paper addresses these issues and proposes a novel generic approach with framework to protect and preserve the user's privacy. Future work should be there focusing on improving the algorithm, policy and authorization strategies in dynamic real time cloud environment to adapt its practicability without effecting the performance of cloud computing.

Acknowledgements

This work is a part of the Research Project sponsored under the scheme of Major Research Project, UGC, India. Reference number F.No. 41-616/2012 (SR), dated 18 July 2012. The authors would like to express their thanks for the financial support offered by the Sponsored Agency.

References

Al-Muhtadia, Jalal, Hillb, Raquel, Al-Rwaisa, Sumayah, 2011. Access control using threshold cryptography for ubiquitous computing environments. J. King Saud Univ. – Comput. Inf. Sci. 23 (2), 71–78.

Chadwick, David W., Fatema, Kaniz, 2012. A privacy preserving authorization system for the cloud. Elsevier-J. Comput. Syst. Sci., 1359–1373

Chandramohan, D., Vengattaraman, T., Basha, M.S.S., Dhavachelvan, P., 2012. MSRCC–Mitigation of security risks in cloud computing, Springer Book Series-AISC-2012, India, vol. 176, doi: http://dx.doi.org/10.1007/978-3-642-31513-8_54, ISBN: 978-3-642-31513-8, pp. 525–532.

Chandramohan, D, Vengattaraman, T., Dhavachelvan, P., 2012. HPPC-hierarchical Petri-net based privacy nominal model approach for Cloud. In: Annual IEEE India Conference (INDICON) Kochi, 2012, 1052, doi: http://dx.doi.org/10.1109/INDCON.2012.6420771, ISBN: 978-1-4673-2270-6. pp. 1047–1052.

Chandramohan, D., Vengattaraman, T., Rajaguru, D., Baskaran, R., Dhavachelvan, P., "A privacy preserving representation for web service communicators' in the cloud. In" QSHINE-2013, 9th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, India, Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, vol. 115, doi: http://dx.doi.org/10.1007/978-3-642-37949-9_44, ISBN:978-3-642-37948-2, pp. 496–506.

Debnath, Ashmita, Singaravelu, Pradheepkumar, Verma, Shekhar, 2014. Privacy in wireless sensor networks using ring signature. J. King Saud Univ. – Comput. Inf. Sci. 26 (2), 228–236.

Dhasarathan, C., Thirumal, V., Ponnurangam, D., 2015. Data privacy breach prevention framework for the cloud service. Security Comm. Networks 8, 982–1005. http://dx.doi.org/10.1002/sec.1054.

Dropbox confirms it got hacked, will offer two-factor authentication Spammers used stolen password to access list of Dropbox user e-mails.Aug-12, News-Conde Nast <http://www.Foxnews.com>.

Facebook Vows to Fix Major Privacy Breach-Australian Report-Sep-2011, <http://www.Foxnews.com>.

Google to pay $22.5 million to settle privacy charges: July-2012 WSJ-Wall Street Journal, <http://www.IBM Live>.

Hao, Zhuo, Zhong, Sheng, Yu, Nenghai, 2011. A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability. IEEE Trans. Knowl. Data Eng. 23 (9), 1432–1437.

Huang, Ruwei, Yu, Si, Zhuang, Wei, Gui, Xiaolin, November 2010. Design of privacy-preserving cloud storage framework, In: The Ninth International Conference on Grid and Cloud Computing, Nanjing, Jiangsu, China, pp. 1–5.

Huang, RuWei, Gui, XiaoLin, Yu, Si, Zhuang, Wei, 2011. Research on privacy-preserving cloud storage framework supporting ciphertext retrieval. International Conference on Network Computing and Information Security (NCIS) 1, 93–97.

Itani, Wassim, Kayssi, Ayman, Chehab, Ali, 2009. Privacy as a service-privacy-aware data storage and processing in cloud computing architectures. In: Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, pp. 711–716.

Khan, S.M., Hamlen, K.W., 2012. Anonymous cloud: a data ownership privacy provider framework in cloud computing. In: IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 170–176.

Li, Jun, Stephenson, Bryan, Motahari-Nezhad, Hamid Reza, Singhal, Sharad, 2011. IEEE Trans. Serv. Comput. 4 (4), 340–354.

Lin, Huang, Shao, Jun, Zhang, Chi, Fang, Yuguang, 2013. CAM: cloud-assisted privacy preserving mobile health monitoring. In: IEEE Transactions on Information Forensics and Security, vol. 8 (6), pp. 985–997.

Linkedln sheds more light on privacy breach, San-fancisco: LinkedIn Corp, criticized for inadequate network security after hackers exposed millions of its users' passwords Jun-2012, <http://www.IBM Live>.

Liu, Qin, Wang, Guojun, Wu, Jie, 2009. An efficient privacy preserving keyword search scheme in cloud computing. In: IEEE International Conference on Computational Science and Engineering, CSE '09, pp. 715–720.

Liu, Qin, Wang, Guojun, Jie, Wu, 2012. Secure and privacy preserving keyword searching for cloud storage services. Elsevier-J. Network Comput. Appl. 35 (3), 927–933.

Nimgaonkar, S., Kotikela, S., Gomathisankaran, M., 2012. CTrust: a framework for secure and trustworthy application execution in cloud computing. In: International Conference on Cyber Security (CyberSecurity), pp. 24–31.

Pearson, Siani, 2009. Taking account of privacy when designing cloud computing services. In: IEEE ICSE Workshop on Software Engineering Challenges of Cloud Computing, CLOUD, pp. 44–52.

Ray, Sangram, Biswas, G.P., 2014. A certificate authority (CA)-based cryptographic solution for HIPAA privacy/security regulations. J. King Saud Univ. – Comput. Inf. Sci. 26 (2), 170–180.

Singhal, M., Chandrasekhar, S., Ge, Tingjian, Sandhu, R., Krishnan, R., Ahn, Gail-Joon, Bertino, E., 2013. Collaboration in multicloud computing environments: framework and security issues. IEEE Comput. Mag. 46 (2), 76–84.

Wang, Jian, Zhao, Yan, Jiang, Shuo, Le, Jiajin, 2010. Providing privacy preserving in cloud computing. In: 3rd IEEE Conference on Human System Interactions (HSI), pp. 472–475.

Wang, Guojun, Liu, Qin, Wu, Jie, Guo, Minyi, 2011. Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers. Comput. Secur. – Elsevier, 320–331.

Wang, Qian, Wang, Cong, Ren, Kui, Lou, Wenjing, Li, Jin, 2011. Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans. Parallel Distrib. Syst. 22 (5), 847–859.

Wang, Q., Zeng, W., Tian, J., 2014. A compressive sensing based secure watermark detection and privacy preserving storage framework. IEEE Trans. Image Process. 23 (3).

Wei, Yang, Jianpeng, Zhao, Junmao, Zhu, Wei, Zhong, Xinlei, Yao, 2012. Design and implementation of security cloud storage framework. In: Second International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC), pp. 323–326.

<http://www.zdnet.com/blog/berlind/phishing-based-breach-of-salesforce>.

Zhang, Gaofeng, Yang, Yun, Chen, Jinjun, 2012. A historical probability based noise generation strategy for privacy protection in cloud computing. Elsevier-J. Comput. Syst. Sci. 78 (5), 1374–1381.

Zhang, Joy Ying, Wu, Pang, Zhu, Jiang, Hu, Hao, Bonomi, Flavio, 2013. Privacy-preserved mobile sensing through hybrid cloud trust framework. In: IEEE Sixth International Conference on Cloud Computing (CLOUD), pp. 952–953.

Zhao, Gansen, Li, Ziliu, Li, Wenjun, Zhang, Hao, Tang, Yong, 2012. Privacy enhancing framework on PaaS. In: International Conference on Cloud and Service Computing (CSC), pp. 131–137.

Zhou, Minqi, Zhang, Rong, Xie, Wei, Qian, Weining, Zhou, Aoying, 2010. Security and privacy in cloud computing a survey. In: Sixth IEEE International Conference on Semantics Knowledge and Grid (SKG), pp. 105–112.