Privacy Protection And Collusion Avoidance Solution For Cloud Computing Users

Maria M. Abur, Sahalu B. Junaidu, Afolayan A. Obiniyi and Saleh E. Abdullahi
Department of Computer Science,
Ahmadu Bello University, Zaria, Nigeria

mmabur1@gmail.com, aaobiniyi@gmail.com and seabdullahi@yahoo.com

Abstract - *Privacy is one of the security issues affecting Cloud Computing Users. It is about securing the Personal Identifiable Information (PII) or Users' attributes on the cloud. Although researches for addressing privacy on the cloud have been carried out, users' PII still remain vulnerable as existing researches lack efficient control of user's attributes of sensitive data on the cloud. Similarly, users are vulnerable to malicious Service Providers (SPs) that may collude to profile a user's identity in a cloud environment. This paper has presented an Enhanced Privacy Protection solution for the control of attributes on the Identity Provider (IdP). Also, it has established an algorithm for the effective dissemination of user's attribute between the Identity Provider and Service Providers, capable of ensuring transparency in transactions thus preserving user's privacy. Moreover, this research has demonstrated the introduction and use of a mechanism called Privacy Token (PT) (an extra security layer) that auto generates all attribute fields (i.e. pseudonyms) as required by each Service Provider (SP) in a secure-manner including trust relationship between the user and the receiving SP without exposing real identities of the user, accompanied with Timestamp and thus avoiding any compromise of the User's privacy. Thereby, aiding proper dissemination of Users' attributes between the IdP and the SPs. Furthermore, preventing SPs from maliciously conspiring (i.e. colluding) to assemble users' attributes on the Cloud environment without users' awareness of it. The correct operations of the research conceptual ideas were verified through a prototype developed with Java Programming Language.*

Keywords: **Cloud Computing, Security, User's attributes, Service Provider and Privacy Token**

I. INTRODUCTION

Cloud computing is Cyberspace computing, where systems, packages, data and other required services are dispensed. It is based on pay before accessing services involved in distributing hosted facilities over the Web. Cloud computing has generated a very significant interest in educational, industrial and businesses setups due to its many gains [1] – [3]. However, Cloud computing is in its early stage of development and is faced with many difficulties [1] – [3]. Researches in ([4], [7], [8] and [9]) have shown that security issues are the major concerns that have prevented the wide adoption of cloud computing. One of the security issues is privacy which is about securing the users 'attributes on the cloud ([10], [11], [12], [13] and [15]). Although researches for addressing privacy on the cloud exist: However, users' PII remain vulnerable as existing researches require enhancements to be effective and efficient.

The general problem of Cloud computing is: Privacy, Performance and Interoperability. Privacy issues include: Lack of control of User's attribute, data breaches, leaks and loss of data. uApprove, uApprove.JP and Temple Data Dissemination (TDD) were used in addressing these challenges, [10, 13 &14]. Despite all these solutions, the cloud is still without adequate protection. Users are endangered to malicious Service Providers (SPs) that may connive to expose a user's identity in a cloud atmosphere [15].

II. EXISTING SYSTEM

On the existing system, users enter and encrypt their attributes themselves on the Identity Provider (IdP) either using Public key which will automatically generate the corresponding private key for the user or using passphrase (sentence) a user must provide to decrypt encrypted data before sending to an SP needing it to release resources to the user, [10]. These keys can easily be hijacked (or copied) or even stolen by anybody who seizes the users' system, ([4], [5] and [6]). Secondly, if the user sends His generated key to his email for future usage it can easily be spammed, sniffed or spoofed or even attacked by Man in the Middle (MIM) and Denial of Service (DOS) on the network, [4] and [6]. These issues are as a result of data leakage.

Similarly, browsers are faced with the data leakage issues of which the browsers used by users on the existing system are not an exception as: prevalence of content-controlled code, flexibility of the JavaScript runtime, lack of systems programming primitives needed to implement crypto and crushing weight of the installed base of users are problems. Each of these issues creates security gaps that are fatal to secure crypto (i.e. leading to data leakage). Attackers may exploit them to defeat systems that should otherwise be secure. There may be no way to address them without fixing browsers, [16] and [17].

Although, researchers so far, have worked on securing the privacy of users' attributes on the IdPs end, though there is still data leakage, as User's attributes can easily be spammed, sniffed or spoofed or even attacked by man in the middle (MIM) and denial of service on the network before reaching the destination, hence compromising the privacy of the user, [4] and [6]. On the other end (i.e. from IdP to SP) is not secured, [12]. As, there are still issues to be dealt with on the SP side; this paper has provided control on the SP furthermore, preventing collusion that may occur due to malicious activities in the cloud that causes users harm.

III. THE PROTOTYPE SYSTEM SOLUTION.

This paper introduces the following solutions to secure the privacy of the user on the Cloud:

a) Develop PII privacy model for the effective control of users 'attributes on the IdP.

b) Develop an algorithm to provide support for the dissemination of user's data on the IdP and SP.

c) Incorporate a mechanism on the proposed model to prevent collusion among multiple SPs.

d) The implementation of the Prototype system.

A. Develop PII privacy model for the effective control of users 'attributes on the IdP.

This is an improvement on the existing PII privacy model. Users would enter their attributes as plaintext into identity provider (IdP). This is to complement for the weaknesses of the existing system already discussed in section two of this paper. Users attributes are entered as plaintext and secured by the IdP using two security measures: Advanced Encryption Standard (AES)-128 cryptography (to convert the User's attributes into ciphertext) and then using Discrete Cosine Transform Modulus three (DCT-M3) steganography algorithm which uses modulus 3 as a base factor (to hide the ciphertext inside a cover image given rise to a stego-image) as it goes through the network to be kept on the IdP. This is as a result of the fact that each of these security measures has its own peculiar advantages. However, by joining both of them will provide a stronger security solution.

Furthermore, the DCT-M3 steganography technique removes any doubt or suspicion to detect the attributes on the cyphertext from the stego-image. For a hacker (intruder) to get through he must first need to defeat the Steganography technique which is hard and then breakthrough the AES-128 cryptographic technique to decode the encrypted (user's attributes) or PII which is extremely difficult. The processes involved in the proposed PII privacy model for the effective control of users' attribute is indicated in Fig. 1.

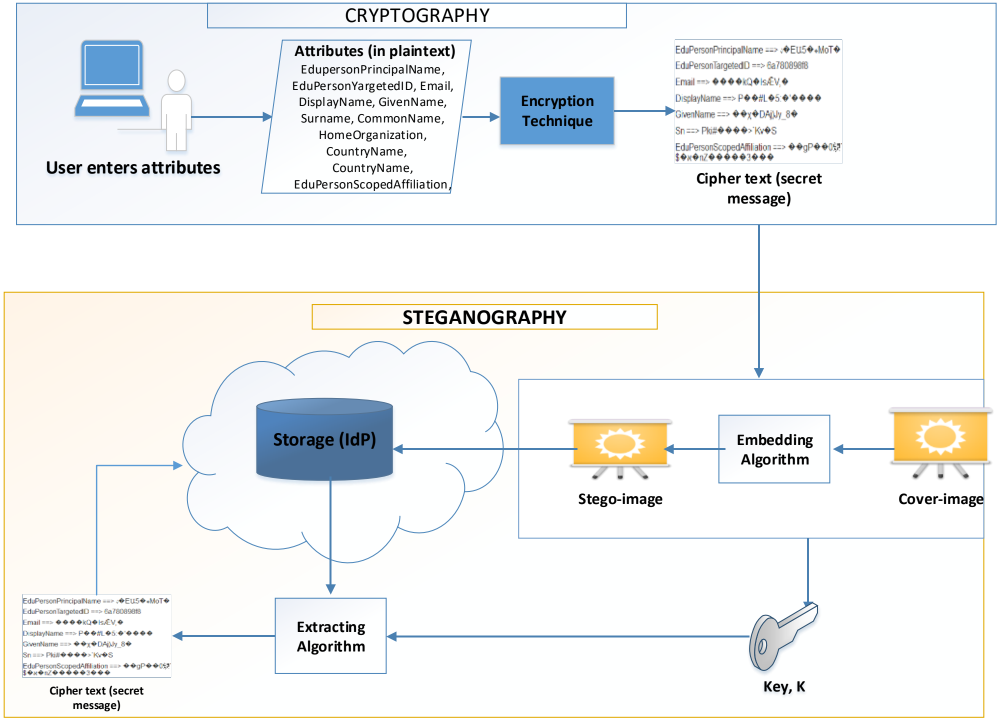

CRYPTOGRAPHY
User enters attributes
Attributes (in plaintext)
EdupersonPrincipalName, EduPersonYargetedID, Email, DisplayName, GivenName, Surname, CommonName, HomeOrganization, CountryName, CountryName, EduPersonScopedAffiliation,
Encryption Technique
EduPersonPrincipalName ==>
EduPersonTargetedID ==> 6a780898f8
Email ==>
DisplayName ==>
GivenName ==>
Sn ==>
EduPersonScopedAffiliation ==>
Cipher text (secret message)
STEGANOGRAPHY
Storage (IdP)
Stego-image
Embedding Algorithm
Cover-image
Extracting Algorithm
Key, K
Cipher text (secret message)

Fig. 1: Prototype PII Privacy Model

Similarly, the procedure for the message (User's attribute) hiding i.e. using (AES-128 + DCT-M3) and the respective extraction stage as illustrated on Figure 2 are described in sections three (1) and (2) respectively.

1. The attribute hiding stage can be summarized as:

    i. The attributes (secret message) are first received as plaintext and converted to cyphertext by encrypting with the AES-128 encryption algorithm.

    ii. Obtain the binary representation of the cyphertext.

    iii. Cover image is selected and then switch the RGB colour layers of the cover image into three different components (Y, Cb and Cr).

    iv. After that, translate the image into transform domain by transforming the pixel data into 8*8 block DCT coefficients using equation (1).

$$F\ (u,\ v) = \frac{1}{4}C(u)C(v) \sum_{x=0}^{7}\sum_{y=0}^{7} f\ (x,\ y) * \cos\left[\frac{(2x+1)u\pi}{16}\right] \times \cos\left[\frac{(2y+1)v\pi}{16}\right] \qquad \textbf{A.1}$$

    v. Generate a randomized sequence with secret key, K using pseudo random method.

    vi. Select a fixed place of two DCT coefficients which will be altered to implant the cipher (thereby avoiding the DC component of each DCT coefficients block).

    vii. Inside each block of 64 coefficients implant only two bits (Pair) as follows:

    - Compute the difference between non-overlapping pair of AC coefficients which are selected before and then.
    - Change DCT coefficients values based on the original values and the message bits accordingly.

    viii. Complete the implanting until the message bit stream is finished.

    ix. Restore original sequence of the DCT blocks using the key, K.

    x. Quantize the image using a quantization table.

    xi. Re-order the values using Zig-Zag ordering.

    xii. Use Huffman lossless compression coding to compress the image.

2. The extracting stage can be summarized as:

    i. Convert the stego-image into transform domain by transforming the pixel data into 8*8 block DCT coefficients.

    ii. Generate randomized sequence with key, K using pseudo random method.

    iii. Within each block of 64 coefficients extract two bits (Pair).

    iv. Concatenate the extracted sub message pairs to get a stream of bits.

    v. Uncompressing the stream of bits to get the original message (cyphertext).

B. Develop an algorithm to provide support for the dissemination of user's data on the IdP and SP.

The prototype algorithm for the effective dissemination of user's attribute between the IdP and SP is illustrated in Fig. 2. The algorithm is to aid meeting the needs of cloud users by preserving their privacy during dissemination of finding secure and reliable services and thereby avoiding data leakage and not exposing their attributes for any dubious intention.

```
Algorithm One (1): Dissemination Algorithm between IdP and SP
Input: Consent page
Output: Tokens
Begin
    While (user selects Service Provider (SP)) do
          Displays SP's terms of usage and consent page to user.
        If (user agrees) Then
          IdP generates and sends Token to SP and resource release to User to accesses
        Else
          Token is not generated
        End If
    End While
End
```

Fig. 2: Prototype Dissemination Algorithm between IdP and SP

C. *Incorporate a mechanism on the proposed model to prevent collusion among multiple SPs.*

The prototype model introduces use of Tokens generated and disseminated by IdP to SPs on behalf of users to release resources to them. The Token is adding an extra security layer that auto generate all attribute fields as required by each SP without compromising the privacy of the user. By introducing and using the following Token policies: Tokens must not carry any original attribute about a user. Generated Tokens of a user to SPs must be different (i.e. unique to each SP) in any case and used only once (i.e. one time usage) within the stipulated timestamp. IdP is acting as a guarantor for the user and is responsible for doing all verifications on behalf of the user. This way the user can access resources on the SP. Hence users' original attributes cannot be kept with SPs. Privacy Tokens ensure transparency in transactions thus protecting user's privacy. The Privacy Tokens has been implemented with Java programming language in this paper. Considering Table 1 with list of Service Providers (SPs), Resources and Attributes required by these SPs to release resources to the users.

TABLE 1:
LIST OF SERVICE PROVIDERS (SPS), SERVICES AND ATTRIBUTES REQUIRED.

Service Providers (SPs)	**Needed Attributes to be release (H_i)**	**Resources to Access by Users (R_i)**
SP1	Principal Name, Targeted ID, Email, Display Name, Given Name, Surname and Scoped Affiliation	R1 & R2
SP2	Targeted ID, Common Name, Principal Name, Email, Display Name, Scoped Affiliation Home Organization & Home Organization Type	R3
SP3	Targeted ID, Assurance, email, display Name, common name and/or given Name, surname, Scoped Affiliation, organization Unit, Organization Acronym, Country Code, country Name & Home Organization	R4 & R5
SP4	Email, Display/Common name, Affiliation Scoped affiliation, Principal Name Persistent/Targeted Identifier, SCHAC home organization SCHAC home organization type	R6
SP5	Entitlement and Scoped Affiliation.	R7

Assuming ***G*** represents entity IdP and $K_i = (K_1, K_2 \ldots K_n)$ represent an entity ***SP,*** $H_i = (H_1, H_2, \ldots, H_n)$ is user's attributes in plaintext, **H_i'** is the attributes needed to be sent to the SP that request for them and **R_i** represent Resources to be release to a user.

If ***G*** takes an input **H_i** from a user and sends **H_i'** to the entity **K_i** based on the user's consent, equations **C.1** is formed.

$$G{:}Hi \longrightarrow H_i' \xrightarrow[W_i]{} K_i \qquad \mathbf{C.1}$$

After which, **K_i** uses **H_i'** to release Resources **R_i**, to the user, which gives rise to equation **C.2**

$$K_i{:}H'_i \xrightarrow[W_i]{} R_i \quad \text{to a user, } \mathbf{U} \qquad \mathbf{C.2}$$

Equations **C.1** and **C.2** represent the mathematical model of the existing system. It follows that there may be possibility for data leakage problems $W_{1\text{-}4}$, to occur during communication between **G** and **K_i** on equation **C.1** where W_1 is Hacking, W_2 is Sniffing, W_3 is malicious insider attacks and W_4 is Collusion.

Now, let **T_i** be introduced to handle the shortcoming of the existing system, where

$T_i = (T_1, T_2, \ldots, T_n)$ represents Privacy Token.

Equations **C.3** and **C.4** represents the prototype system for aiding dissemination of user's attributes and tackling data leakage.

G takes user's input H_i with user's consent ***E***, and then sends T_i to K_i where K_i uses T_i to release resource R_i to the user ***U***. Given rise to equations **C.3** and **C.4** respectively.

$$G{:}\ H_i\ E \rightarrow T_i \rightarrow K_i \qquad \mathbf{C.3}$$

$$K_i{:}\ T_i \rightarrow R_i \text{ to a user, } U \qquad \mathbf{C.4}$$

Hence, T_i cannot be reused even if it was kept for future usage by K_i. This is due to the fact that T_i does not carry any original attribute of the user but Pseudonyms and it expires at a given timestamp. It is only a means by which K_i can be used to grant user, **U** access to resource, R_i.

Since Privacy Token T_1 in the hand of Service Provider H_1 received from an IdP, ***G*** for a user U_i is not the same with what Service Provider H_2 receives as Privacy Token from ***G***. Hence, the potential Collusion problem is tackled and User's Privacy is preserved.

The architecture and Workflow of the Prototype System is represented on Figs. 4 & 5 respectively.

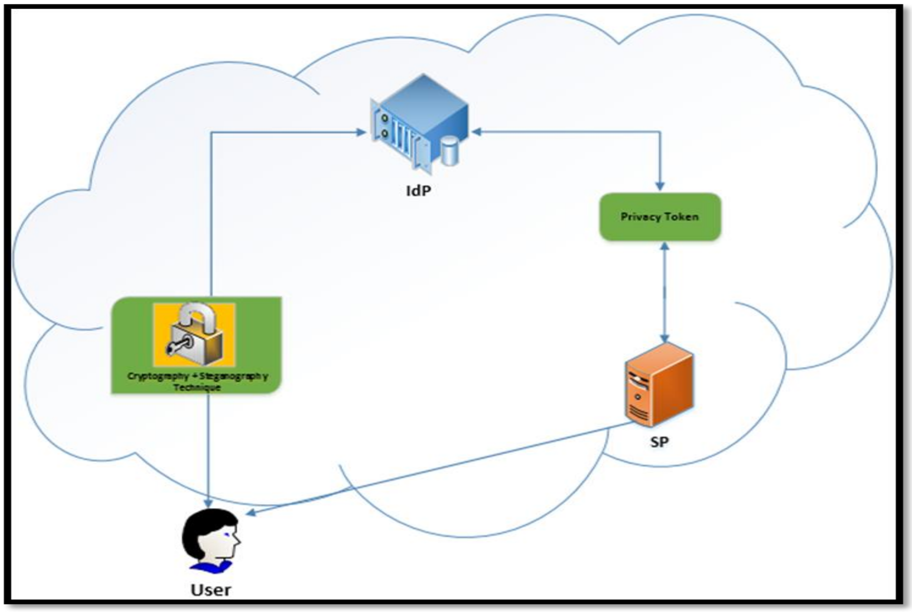
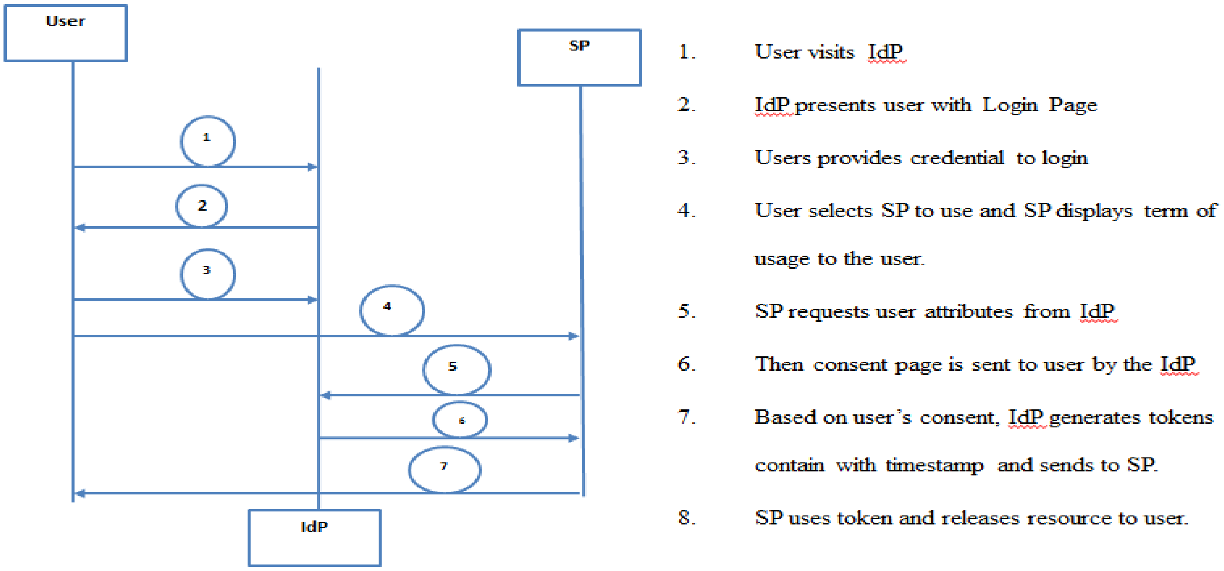


Fig. 4: Architecture of the Prototype System



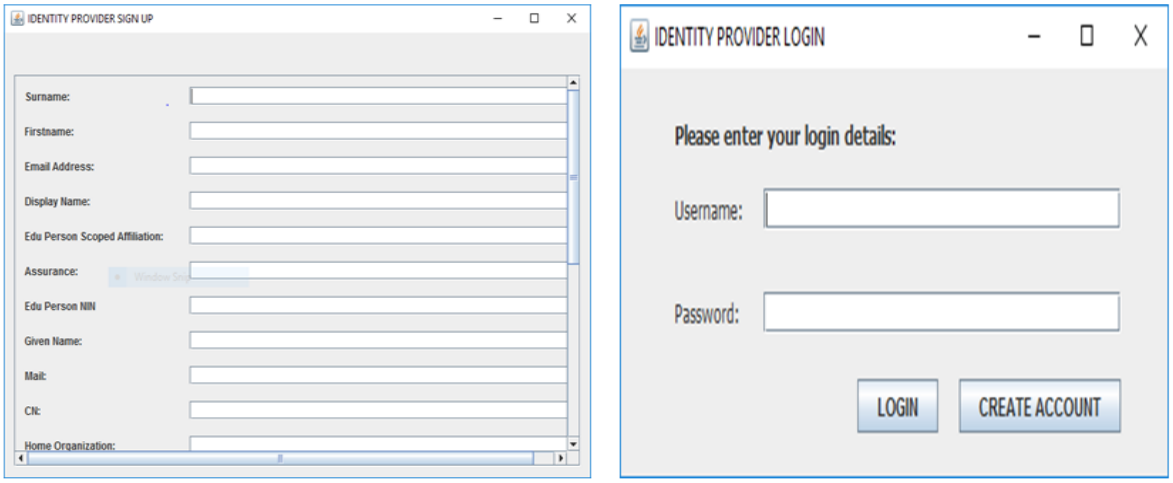
1. User visits IdP
2. IdP presents user with Login Page
3. Users provides credential to login
4. User selects SP to use and SP displays term of usage to the user.
5. SP requests user attributes from IdP
6. Then consent page is sent to user by the IdP
7. Based on user's consent, IdP generates tokens contain with timestamp and sends to SP.
8. SP uses token and releases resource to user.

Fig. 5: Prototype System Workflow

D. Implementation of the prototype system.

The program developed incorporates the entities mainly involved in the Cloud Computing environment (i.e. federated environment) namely: the attributes, user, Identity provider and Service Provider. A layer of security was introduced at the Identity provider to enhance user's privacy and then the Privacy Token (PT) was also introduced to ensure that no original attributes of a user was sent to a Service provider for further protection of the user's privacy during dissemination on the Cloud. It is also mandated for a Cloud user to have an account with the Identity Provider, so that all the attributes required for various services needed by the user are properly stored using the AES-128 and DCT-M3 Steganography techniques on the Identity Provider. The program used a single login key structure so that there is minimal delay in its operation. After logging on to the IdP, the user is presented with a list of service providers he/she can access then he/she can select the service(s) he/she wants to access from the service provider (SP). Subsequently, the SP presents the user with it Term of Usage (ToU) and the IdP presents the user with consent page on how their attributes is to be disseminated and if the user agrees the PT is generated and sent to the SP in order to release resources to the user. This way, adding an extra layer of security of the user's attributes hence preserving the privacy of the user.

The Identity Provider Sign up Page and the Login page are illustrated on Fig. 6 (i) & (ii) respectively.


IDENTITY PROVIDER SIGN UP

Surname:

Firstname:

Email Address:

Display Name:

Edu Person Scoped Affiliation:

Assurance:

Edu Person NIN

Given Name:

Mail:

CN:

Home Organization:

IDENTITY PROVIDER LOGIN

Please enter your login details:

Username:

Password:

Fig. 6 (i): Identity Provider Sign up Page. (ii) Login page

Similarly, Fig. 7 shows the list of Service Providers considered in the research case study.

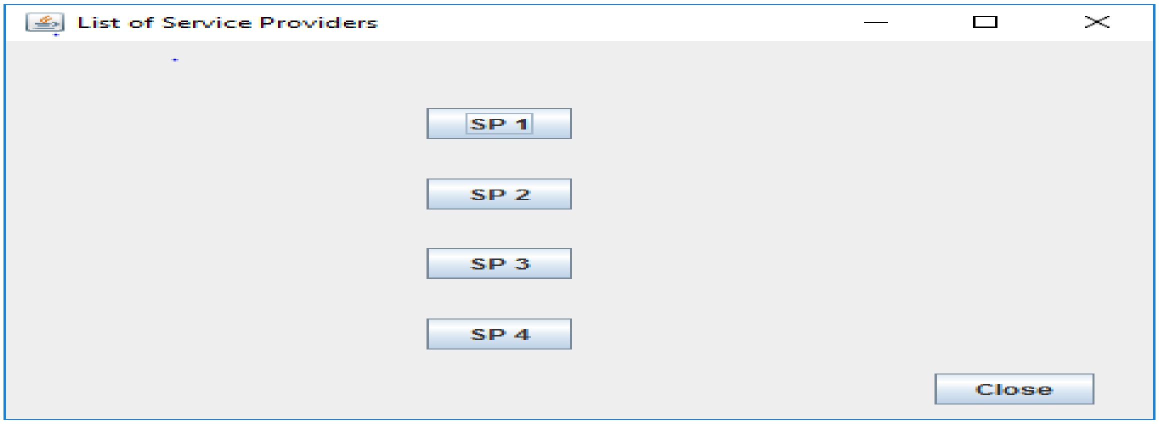
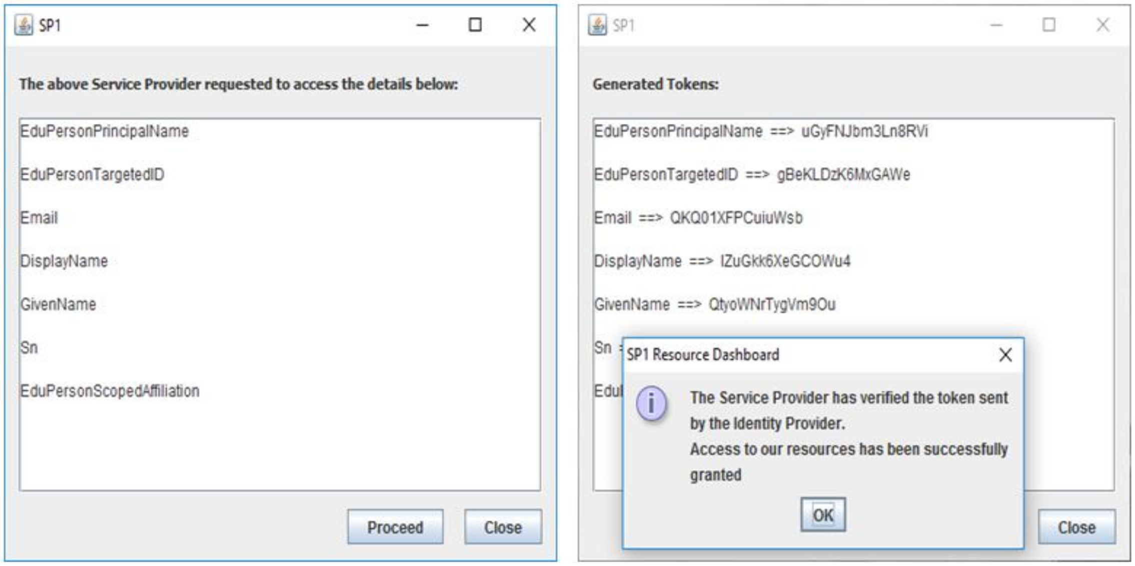

List of Service Providers

Figure 7: List of Service Providers

Moreover, Figs 8 & 9 demonstrates (i): List of Attributes required by SP1 & SP2. (ii) Generated Tokens sent to SP1 & SP2 on behalf of a user respectively.


SP1

The above Service Provider requested to access the details below:

```
EduPersonPrincipalName
EduPersonTargetedID
Email
DisplayName
GivenName
Sn
EduPersonScopedAffiliation
```

SP1

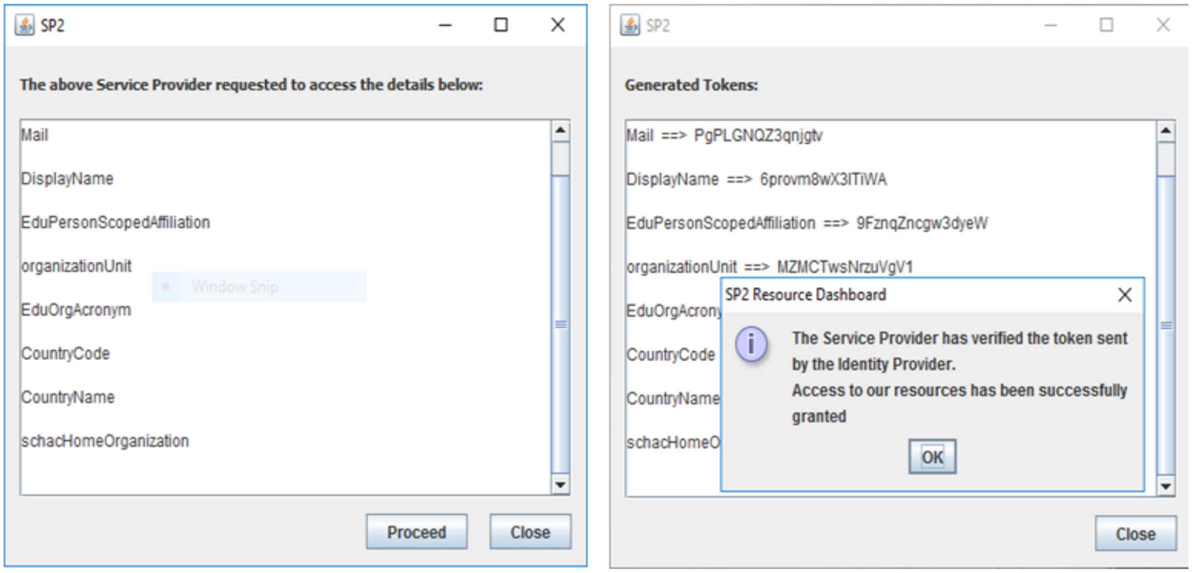
Generated Tokens:

```
EduPersonPrincipalName ==> uGyFNJbm3Ln8RVi
EduPersonTargetedID ==> gBeKLDzK6MxGAWe
Email ==> QKQ01XFPCuiuWsb
DisplayName ==> IZuGkk6XeGCOWu4
GivenName ==> QtyoWNrTygVm9Ou
Sn =
Edu
```

SP1 Resource Dashboard

The Service Provider has verified the token sent by the Identity Provider.
Access to our resources has been successfully granted

Fig. 8 (i): List of Attributes required by SP1. (ii) Generated Tokens sent to SP1 on behalf of a user



SP2

The above Service Provider requested to access the details below:

Mail

DisplayName

EduPersonScopedAffiliation

organizationUnit

EduOrgAcronym

CountryCode

CountryName

schacHomeOrganization

SP2

Generated Tokens:

Mail ==> PgPLGNQZ3qnjgtv

DisplayName ==> 6provm8wX3lTiWA

EduPersonScopedAffiliation ==> 9FznqZncgw3dyeW

organizationUnit ==> MZMCTwsNrzuVgV1

EduOrgAcrony

CountryCode

CountryName

schacHomeO

SP2 Resource Dashboard

The Service Provider has verified the token sent by the Identity Provider.
Access to our resources has been successfully granted

Fig. 9 (i): List of Attributes required by SP2. (ii) Generated Tokens sent to SP2 on behalf of a user

IV. CONCLUSION

This paper has presented an Enhanced Privacy Protection solution for the control of attributes on the Identity Provider (IdP). Also, it has established an algorithm for the effective dissemination of user's attribute between the Identity Provider and Service Providers, capable of ensuring transparency in transactions thus preserving user's privacy. Moreover, this paper has demonstrated the introduction and use of a mechanism called Privacy Token (PT) (an extra security layer) that auto generates all attribute fields (i.e. pseudonyms) as required by each Service Provider (SP) in a secure manner including trust relationship between the user and the receiving SP without exposing real identities of the user, accompanied with Timestamp and thus avoiding any compromise of the User's privacy. Thereby, aiding proper dissemination of Users' attributes between the IdP and the SPs for the release of resources to users. This, as a result, will prevent the SPs from keeping user's attributes and further preventing them from maliciously conspiring to assemble users' attributes on the Cloud.

REFERENCES

[1] M. M. Abur, O. S. Adewale, S.B. Junaidu, (2015): Cloud Computing Challenges: A review on Security and Privacy issues. Proceedings of the ACM International Conference on Computer Science Research and Innovations (CoSRI), Ibadan pp. 89-92.

[2] A.A.C. Fauzi, A. Noraziah, T. Herawan, and M. N. Zin, 2012, March. On Cloud Computing security issues. In Asian Conference on Intelligent Information and Database Systems (pp. 560-569). Springer Berlin Heidelberg.

[3] N. Khan, A. Noraziah, T. Herawan, and Deris, M. M., 2012c, September. Cloud Computing: analysis of various services. In International Conference on Information Computing and Applications (pp. 397-404). Springer Berlin Heidelberg.

[4] P. N. Asha, T. Mahalakshmi, S. Archana, S. C. Lingareddy, (2016): Wireless Sensor Networks: A Survey on Security Threats Issues and Challenges. International Journal of Computer Science and Mobile Computing, Vol.5 Issue.5, pg. 249-26.

[5] Cloud Security Alliance (CSA). 2013: The Nine Notorious Threats. Top threats working group.

[6] Cloud Security Alliance (CSA). 2016: The Treacherous 12 - Cloud Computing Top Threats

[7] B. P. Rima, E. Choi, & I. Lumb, (2009): A Taxonomy and Survey of Cloud Computing Systems. In *Proc. of the* 5^{Th} *International Joint Conference on INCIMS and IDC, NCM '09*, IEEE Press, 44-51.

[8] M. Zhou, R. Zhang R, W. Xie, W. Quian & A. Zhou, (2010) Security and Privacy in cloud: Survey. In *Proc. Of the 6Th International Conference on Semantics, Knowledge and Grids*, IEEE Press, 105-112.

[9] D. Chen & H. Zhao (2012): Data Security and Privacy Protection Issues in Cloud Computing. In *Proc. of the* 1^{st} *International conference on Computer Science and Electronics Engineering*, 647-651.

[10] R. Weingartner, (2014) "Dissemination control of data Sensitive environment in Federated Systems". *M.Sc. Computer Science Thesis*, Department of Informatics and Statistics, Federal University of Santa Catarina, Brazil.

[11] S. Betg´e-Brezetz, G. B. Kamga, M. Ghorbel, and M. P. Dupont, "Privacy control in the cloud based on multilevel policy enforcement,"

In *Proceedings of 2012 IEEE 1st International Conference on Cloud Networking* (CLOUDNET), IEEE Press 2012, 167–169.

[12] S. Betge-Brezetz, G. B. Kamga, M. P. Dupont & A. Guesmi (2013). End-to-end Privacy Policy Enforcement in Cloud Infrastructure," In *Proceedings of IEEE 2nd International Conference Cloud Networking* (CloudNet 2013), 25-32.

[13] T. Orawiwattanakul, K. Yamaji, M. Nakamura, T. Kataoka & N. Sonehara (2010): "User-controlled privacy protection with attribute-filther mechanism for a federated SSO environment using Shibboleth," in *IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (3PGCIC), IEEE Press, 243-249.

[14] SWITCH, (2014). "*uapprove - user consent module for shibboleth identity providers*," retrieved: [Online]. Available: https://www.switch.ch/aai/support/tools/uApprove.html

[15] M. M. Abur, S.B. Junaidu, S. Danjuma, S. Arlis, R. Ritonga, T. Herawan (2018): Towards a Privacy Mechanism for Preventing Malicious Collusion of Multiple Service Providers (SPs) on the Cloud. In: Bhateja V., Nguyen B., Nguyen N., Satapathy S., Le DN. (eds) Information Systems Design and Intelligent Applications. Advances in Intelligent Systems and Computing, Singapore: Springer, vol 672.

[16] T. Ptacek, (2011): Javascript Cryptography Considered Harmful, What Do You Mean, "Javascript Cryptography"?https://www.nccgroup.trust/us/about-us/newsroom-and- events/blog/2011/august/javascript-cryptography-considered-harmful/22-08-2017.

[17] S. Lekies, B. Stock, M. Wentzel (2015): The Unexpected Dangers of Dynamic JavaScript 24th USENIX Security Symposium Washington, D.C. pp. 723-735.