Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 89 (2016) 293 – 300

Procedia Computer Science

Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016)

Index Generation and Secure Multi-User Access Control over an Encrypted Cloud Data

S. Raghavendra[a,*], K. Meghana[a], P. A. Doddabasappa[a], C. M. Geeta[a], Rajkumar Buyya[b], K. R. Venugopal[a], S. S. Iyengar[c] and L. M. Patnaik[d]

[a] *University Visvesvaraya College of Engineering, Bangalore, India*
[b] *The University of Melbourne, Australia*
[c] *Florida International University, USA*
[d] *INSA Senior Scientist, National Institute of Advanced Studies, Indian Institute of Science Campus, Bangalore*

Abstract

Cloud computing provides economical and effective solution for sharing data among cloud users with low maintenance cost. The security of data and identity confidentiality while sharing data in multi-owner way cannot be assured by the Cloud Service Providers (CSP's). The Cloud Service Providers are reliable but curious to know the recurrent membership changes in the cloud. In this paper, we propose a secure multi-owner data sharing for dynamic group in the cloud with RSA Chinese Remainder Theorem (RSA-CRT) encryption technique and substring index generation method. RSA-CRT efficiently manages revocation list, key management, with reduced storage and computational overhead. The substring Index generation algorithm reduces the storage space compared to wild card fuzzy alogorithm[1].

Peer-review under responsibility of organizing committee of the Organizing Committee of IMCIP-2016

Keywords: Access Control; Cloud Computing; Key Generation; Keyword Search; Storage and Retrieval.

1. Introduction

Cloud computing is the service which is provided over the Internet. It is used to share resource at low maintenance cost, as service is completely managed by the cloud service vendor. The service is provided on demand and charged as much as the user uses the service. This service is fully managed by the cloud service provider and thus reduce the maintenance complexity, data loss problem as well as reduces capital investment for purchasing hardware and software. This uses a large group of servers which is running at low cost PC technology and provides specialized data processing. Cloud computing is a customer-oriented application in financial portfolios which delivers personalized information to provide data storage and sharing among the members of an organization. It provides a scalable and reliable database which is maintained by the provider. They provide various services such as Communication-as-a-Service (CaaS), Infrastructure-as-a-Service (IaaS), Platform-as-a-Sevice (PaaS), Monitoring-as-a-Service (MaaS) and Software-as-a-Service (SaaS).

*Corresponding author. Tel: +91-9591276777.
E-mail address: raghush86@gmail.com

Peer-review under responsibility of organizing committee of the Organizing Committee of IMCIP-2016
doi:10.1016/j.procs.2016.06.062

In cloud computing, security is provided to the data in cloud by encrypting and uploading file. The data owner gives access to the users by disseminating the decryption key to the authorized users to decrypt the files[2,3]. The early encryption methods includes dynamic broadcast encryption[4] in which the users can be added dynamically and the communicator or broadcaster broadcast the decryption key to the authorized users who in turn can access the files. In this technique, storage overhead increases with the increase in the number of users as well as revoked users.

Due to this drawback many encryption techniques were developed[5–7]. Riedel *et al.*, proposed Plutus[5] which focuses on key management that is highly controllable and scalable. HASBE[6] cipher text attribute based encryption is used to achieve scalability and flexibility in secure access control in which the user's secret key as well as cipher text is associated with the attributes. In a File-Centric Model for Peer-to-Peer File Sharing Systems Li Zou *et al.*,[8] composed a file sharing scheme where the files are organsied into groups and the propagation is studied. These methods have many drawbacks: difficulty in handling the keys when there is increase in the number of group members, revocation list and access control needs to be updating periodically.

Motivation: In MONA[9] data sharing for multi-owner level is considered, where the user can share data with any group member in a secure way. The authorized group users can access the data in the cloud without communicating with the data owners or administrator. The user revocation is achieved without altering the secret (private) key of the user, but the number of sub-keys generated are more for every user of that group. Hence, computation cost increases exponentially with increase in the number of users of the group.

Contribution: The key contributions of our paper are given below:

1. The proposed RSA-CRT scheme is used to share data in a secure manner. The master key updates the authentication of each user in a group; like adding new user to the group or deleting a user from the group while the secret key remains same for the other user in the same group.
2. The substring index generation method reduces the cost of the storage space compared to wild card fuzzy algorithm. The index file is used to retrieve the stored files effectively at low cost.
3. The RSA-CRT scheme security and performance is efficient in key management, updating revocation list and computation cost.

Organization: The paper is organised as follows: Section 2 briefs on related work. Background work is discussed in section 3. Design goals and system model is described in section 4. Detailed analysis of proposed scheme is presented in section 5. Security analysis and performance analysis are discussed in section 6 and section 7 respectively. The conclusions of the paper is contained in section 8.

2. Related Work

Zhou *et al.*,[8] proposed a File-Centric Model for Peer-to-Peer File Sharing Systems. Different behaviors of accessing of files are analyzed, grouped by peers and system is modeled accordingly. The file propagation in the system can be analyzed effectively while downloading and sharing. In this paper, file propagation in low level transition of peer-to-peer model is not considered.

Nafi *et al.*[2] proposed a new user authentication, file encryption and distributed server based cloud computing security architecture that emphasises on information concealment and security. RSA encryption is used for secure communication, Advanced Encryption Standard (AES) for file encryption, MD5 hashing and one time password is used for authentication. The users access files pass through secure channel and one time password for authentication is used in which user enters a new password everytime. Encryption path is transparent to the user and the file is stored securely.

Sun *et al.*,[10] designed an approach towards rebalanced RSA-CRT with Short Public Exponent. This approach speeds up the process of encryption three times faster than the already proposed rebalanced RSA-CRT thus reducing the cost. Public exponent is shortened from 1024 bits down to 512 bits in Rebalanced RSA-CRT.

Wang *et al.*,[11] proposed public auditing system to integrate shared data with efficient user revocation. Proxy re-signature is used to resign the block of the users in case of revocation which helps the user to refrain from downloading their files and $re - signing$ themselves. Deffie-Hellman and Discrete Logarithm is used to define the

security in the system. The files and signing keys data is stored in different servers and the method to secure these information is not proposed in this system.

Bernhard *et al.*,[12] proposed cryptographically enforced permissions for fully decentralized file systems to maintain the integrity of file system and preserve the privacy of files. Symmetric cryptography and digital signature is used for security based on hidden file structure which can be accessed using meta-information. Validity of the file system is checked using integrity verification algorithm and privacy of the file system's content is maintained using a cryptographic data protection scheme and hence, user and group can be accessed.

Many approaches are proposed based on keywords to search and easy retrieval[13–16] of cloud data. Secure ranked multi-keyword search[1] over outsourced encrypted cloud data uses fuzzy algorithm to reduce the index construction time. Further, search time for the keyword can be reduced. Tuo *et al.*,[17] have developed fuzzy keyword search scheme by using partial decryption concept with filter. It improves keyword search at low cost. Partial decryption cannot handle search queries efficiently. Precisely controlled fuzzy keyword search can be considered for improvement.

3. Background Work

The security of the shared data in multi-owner level is considered in MONA: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud[9]. The authenticated members of the group can upload and share data or files with other members of the group in the cloud. The time complexity of file generation, access and deletion is the same irrespective of the number of users in the cloud. When a new user is added or revoked, the existing users private key is not updated. The new users have authority to directly access a file without consulting the group manager. In MONA, users do not have access to multiple data groups in the cloud; only one file can be uploaded to the cloud that increases the computation time and the size of the file which is uploaded must be in mega bytes. The private key of the user is not affected with every revocation but two rounds of encryption and decryption is performed which increases the computation cost. The number of cipher text generated for each user is more and hence increases the key storage size.

3.1 Chinese reminder theorem

Let $u_1, u_2, \ldots, u_r$ be positive integers such that $gcd(u_i, u_j) = 1$ for $i \neq j$. Then linear congruence system is:

$$x \equiv a_1 (\text{mod } u_1); x \equiv a_2 (\text{mod } u_2); \ldots x \equiv a_r (\text{mod } u_r);$$

and has simultaneous solution, and is unique modulo $u_1, u_2, \ldots, u_r$. Example: Find x for the given system of equations:

$$x \equiv 5(\text{mod } 3); x \equiv 3(\text{mod } 11); x \equiv 2(\text{mod } 13);$$

$$u = 3 * 11 * 13 = 429;$$

$$U_1 = 429/3 = 143; U_2 = 429/11 = 39; U_3 = 429/13 = 33$$

Linear Congruence is:

$$143x \equiv 1(\text{mod } 3), 39x \equiv 1(\text{mod } 11), 33x \equiv 1(\text{mod } 13)$$

$$x_1 = 2; x_2 = 2; x_3 = 2; \ \bar{x} = 2.143.5 + 2.39.3 + 2.33.2 = 1796$$

$$\bar{x} = 1796 \equiv x(\text{mod } 429)$$

$$x = 80$$

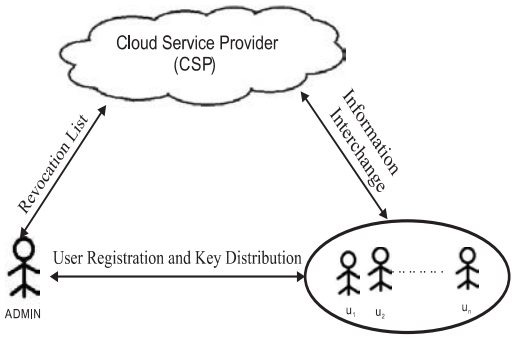

Fig. 1. System Model for User Revocation and Registration.

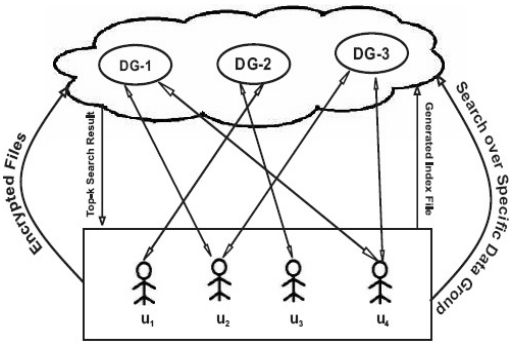

Fig. 2. System Model for Index Generation and Searching.

4. Problem Statement and System Model

4.1 Problem statement

Given a set of n files $(f_1, f_2, f_3, \ldots, f_n)$ that is uploaded on to the cloud with a secure storage and access control for a dynamic set of users, the objectives are:

1. To improve the master key management for access control using RSA-CRT technique and maintain the security of the shared data without updating the secret key of the remaining users.
2. Build index generation for fast retrieval of files with reduce index storage space.

4.2 System model

The System Model is divided into two parts as shown in Figs. 1 and 2. Figure 1 is used for User Registration and Revocation and Fig. 2 represents Index Generation.

In the system model (Fig. 1), a group *admin* registers with the cloud and group members share files and data on the cloud. The members who wish to join the group must register with the group *admin*, the $admin$ authenticates the registed user and the group users can upload, access or modify files on the cloud as depicted in the Fig. 1 and is composed of three main components:

1. *Cloud Service Provider (CSP)*: Cloud is managed by the cloud service provider (CSP) which operates based on pay as you use service. The service provided is considered reliable but inquisitive. The cloud maintains the data and does not modify any uploaded data but might try to check the contents of the data.
2. *Group Admin*: Admin is the one who is trusted and has registered on the cloud. He is responsible for user registration and revocation of misbehaved users. During any conflict, group $admin$ reveals the true identity of the user.
3. *Group Users*: Group users are the members who are authorized to access the cloud. These users can upload and access the files on the cloud and can modify their own files on the cloud.

Figure 2 represents system model for index generation and searching. In Fig. 2, the cloud controls a number of Data Groups (DG). The *admin* authenticates the users. Depending on the permissions given to the users, the users can access, upload or modify the files or data of that particular data group. For instance, consider User-1 (u_1) has given authority to access DG-2 and DG-3 group files as shown in Fig. 2. So u_1 can access or store the encrypted files in DG-2 and DG-3 but does not have permission to modify data of other users of the group.

Each and every file in the cloud has separate index. If the user wants to search for any file it is searched over that data group authorized. Using Top-k search result, the file is searched in groups for which an user is authorized to and the file is retrieved.

1: Select $m_1, \cdots, m_z, n_1, \cdots, n_z$ from a set of prime numbers $m \neq n$
2: **for** k= 1 to z **do**
3: $\chi_i = (m_i - 1) \times (n_i - 1)$;
4: $a_i = (m_i - 1)/2$;
5: $b_i = (n_i - 1)/2$;
6: $e_i = 4 \times$ Integer in RandomNumber + 1;
7: $d_i = e_i^{2(a_i-1)(b_i-1)-1} \bmod 4a_ib_i$;
8: **end for**
9: p = 1;
10: **for** i =1 to z **do**
11: p =$p \times (a_ib_i)$;
12: **end for**
13: **for** i = 1 to z **do**
14: $M[i] = p/(a_ib_i)$;
15: $N[i] = M[i]^{(a_i-1)(b_i-1)-1)} mod(a_ib_i)$;
16: **end for**
17: $e_M = 0$;
18: **for** i = 1 to z **do**
19: $e_M = (e_M + (e_i \times M[i] \times N[i]))mod\, p$;
20: **end for**
21: **while** $(e_M \mod 4! = 1) e_M = e_M + p$; **do**
22: sleep;
23: Interrupt(when j^{th} key pair should be updated)
24: $e_j = 4 \times$Integer in Random + 1;
25: $d_i = e_i^{(2(a_i-1)(b_i-1)-1)} mod 4a_ib_i$;
26: goto 17;
27: **end while**

Algorithm 1. Master Key Generation Algorithm

1: Initialize
2: **for** int i = 0; i < files.length; i = files.next **do**
3: **while** !eof **do**
4: **if** !stopword **then**
5: sub-key = keyword.substring(0,3);
6: bucket = keyword.substring(0,1);
7: enc-sub-key = encryption(sub-key);
8: Goto(bucket)
9: **if** (bucket.contains(enc-sub-key)) && (!bucket.contains(file-id)) **then**
10: frequency ++
11: add to hash map (enc-sub-key, file-id, frequency)
12: else Set frequency = 1;
13: add hash map to bucket (enc-sub-key, file-id, frequency)
14: set frequency = 1;
15: add hash map to bucket (enc-sub-key, file-id, frequency)
16: **end if**
17: **end if**
18: Move to next character
19: **end while**
20: **end for**

Algorithm 2. Index Building Algorithm

5. Proposed Scheme

5.1 Generation of master-key and other keys

In this paper, we emphasize on dynamic access and secure file storage and distribution in cloud for many data groups. We can effectively store our files in the cloud with minimum number of cipher texts and with low time complexity. We focus on security of private keys of each user. Even after a user is revoked or the members data group access is altered, the private keys of other users is unaltered.

5.2 Index building process

The index building algorithm has 2 phases: Initialization and Index Generation phase.

1. *Initialization*: In this phase, substring of the keyword are initialized in bucket. A file-set and sub-keys are generated. The files of file set that have to be uploaded on the cloud are assigned with $file - id$ based on which the files are accessed. The substring of the keyword is generated and assigned to the sub-key that is used during index building process at later stage. The bucket is created and assigned with the first character of the keyword. For instance, the first character present in the bucket name, in which sub-string starting with same alphabet is stored. Encrypted sub-string is assigned to $enc - sub - key$ which is stored in the bucket.
2. *Index Generation*: In this phase, the keyword index is created and stored in their respective buckets. In the index generation phase, the keyword is extracted from the document that has to be uploaded and checked whether it

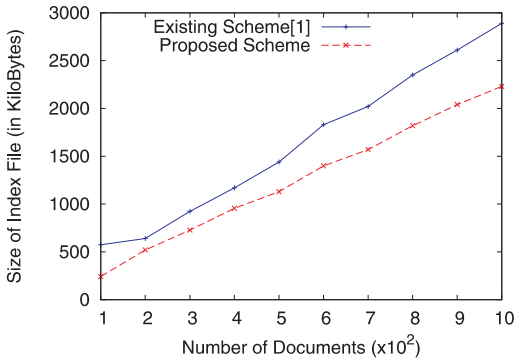

Fig. 3. Size of the Index File.

is a stop word or not. If it is a stop-word, then it is discarded; if not, then further steps of index generation is carried out. To generate the index, the first 3 characters of the keyword are considered, discarding the rest of the characters of the keyword, then it is encrypted and checked with the bucket which is having the encrypted keywords index starting with same character. If the bucket has the entry for that keyword with the same $file-id$, then the frequency count is incremented by 1, else, the new entry is updated and the frequency is assigned with 1, it is repeated for all the keywords of all the files in the file set to be uploaded.

For the above index building algorithm, a list of the files f_1, f_2, f_3, ..., f_n to be uploaded to the cloud is considered. Let us consider the file f_1 which has the words Bangarapet, Bangalore, computer, computing, compute, and come. Another document called f_2 have the words likes computer, computing, Bangladesh, Australia and Austria. Now when this file list is uploaded to the cloud, it first checks the file set length i.e., number of files to be uploaded. Then it starts fetching the files one by one until all the files are indexed and uploaded. To create the index of each file, the following process takes place.

The tabular form of the bucket (b) is shown below:

Keyword (encrypt)	frequency	field-id
Encrypt (ban)	2	f_1
Encrypt (ban)	1	f_2

6. Performance Analysis

The experiment evaluation of the proposed scheme is performed on the real data set: National Science Foundation Research Award Abstracts 1990–2003[18]. This file sets contains a large number of technical keywords and are unique to the files. The entire system is implemented in Java language. The Data owner and the data user use a windows platform with Intel (R) Core (TM) i5-421 M CPU @ 2.60 GHz, 4.00 GB of RAM and commercial public cloud Amazon S3 service to store the Index file and encrypted collection of files.

6.1 Size of the index file

In the existing system, the keywords in the super file do not perform any substring operation. In the proposed system, substring of the given keyword is stored in index file which results in reducing the memory consumed by the file. Hence, the proposed system consumes less space when compared to the existing system; 574 Kb and 243 Kb for 100 files, 1.44 Mb and 1.13 Mb for 500 files, 2.9 Mb and 2.2 Mb for 1000 files space is consumed in the existing and proposed system respectively.

6.2 Security analysis

In cloud computing, many security issues arise such as privacy, data security, confidentiality and authentication. Cloud computing emphasizes for access control and as data security while users shares confidential data via cloud servers, as these are not under the control of trusted environment. Cloud environments are accessed among various individuals and cloud service distributors have exclusive access to the files and content stored in those environments. Hence, confidential data uploaded in a cloud should be kept securely by utilizing various combination of access control, contractual liability and encryption, which offers the benefits of minimum reliance on the cloud service provider.

A fundamental approach for secure data sharing in a cloud setup is to let the data owner encrypt data before outsourcing and issue decryption keys to authorized users. To share the data with a group of users, they need fine-grained data access control in terms of the user (data consumer) and his access privilege to various level of data.

The scheme enables the use of Chinese reminder theorem for public/private key generation and thus prohibits the CSP to automatically re-encrypt the data on each user request that ensures enhanced confidentiality to the data passed to the user. The data owner needs to receive the access request from the user. Therefore, on user revocation, the data owner need not re-encrypt the data and can be offline and thus, scalability is achieved. This approach provides the owner with the privilege of one time encryption and is flexible to allow clients to use different keys to decrypt the data. There is no need to re-generate the key when the users join or leave thus providing easy and flexible way of access control without much key management, and in addition to maintain forward and backward secrecy.

7. Conclusions

In this paper, the concept of Master key generation is used to encrypt and store the contents in the cloud. An efficient index building algorithm is designed for fast and cost efficient file retrieval from the cloud. The Master key generation algorithm where only the master-key is updated with every revocation or membership change, keeping the existing group members private and public keys unaltered. This approach reduces the storage space of the index file and key size. Further this protocol can be enhanced to other form of text and multimedia files. The faster index building algorithm can be explored to retrieval files efficiently.

References

[1] Neelam S. Khan, C. Rama Krishna and Anu Khurana, Secure Ranked Fuzzy Multi-Keyword Search over Outsourced Encrypted Cloud Data, In *IEEE Proceedings of the 2014 International Conference on Computer and Communication Technology (ICCCT)*, pp. 241–249, (2014).

[2] Kawser Wazed Nafi, Tonny Shekha Kar, Sayed Anisul Hoque and MMA Hashem, A Newer User Authentication, File Encryption and Distributed Server Based Cloud Computing Security Architecture, *arXiv preprint arXiv:1303.0598*, (2013).

[3] Cong Wang, Kui Ren and Jia Wang, Secure and Practical Outsourcing of Linear Programming in Cloud Computing, In *IEEE Proceedings of the IEEE INFOCOM, 2011*, pp. 820–828, (2011).

[4] Min-Ho Park, Young-Hoon Park, Han-You Jeong and Seung-Woo Seo, Secure Multiple Multicast Services in Wireless Networks, *IEEE Transactions on Mobile Computing*, (2012).

[5] Mahesh Kallahalla Erik Riedel, Ram Swaminathan Qian Wang and Kevin Fu, Plutus: Scalable Secure File Sharing on Untrusted Storage.

[6] Zhiguo Wan, Jun'e Liu and Robert H Deng, HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing, *IEEE Transactions on Information Forensics and Security*, vol. 7(2), pp. 743–754, (2012).

[7] Giuseppe Ateniese, Kevin Fu, Matthew Green and Susan Hohenberger, Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage, *ACM Transactions on Information and System Security (TISSEC)*, vol. 9(1), pp. 1–30, (2006).

[8] Li Zhou and Mostafa H Ammar, A File-Centric Model for Peer-to-Peer File Sharing Systems, In *IEEE Proceedings of the 11th IEEE International Conference on Network Protocols, 2003*, pp. 28–37, (2003).

[9] Xuefeng Liu, Yuqing Zhang, Boyang Wang and Jingbo Yan, Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud, *IEEE Transactions on Parallel and Distributed Systems*, vol. 24(6), pp. 1182–1191, (2013).

[10] Hung-Min Sun and Mu-En Wu, An Approach Towards Rebalanced RSA-CRT with Short Public Exponent, *IACR Cryptology ePrint Archive*, 2005, pp. 53, (2005).

[11] B Wang, Baochun Li and H Li, Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud, (2014).

[12] Bernhard Amann and Thomas Fuhrmann, Cryptographically Enforced Permissions for Fully Decentralized File Systems, In *IEEE Proceedings of the 2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P)*, pp. 1–10, (2010).

[13] S. Raghavendra, C. M. Geeta, K. Shaila, Rajkumar Buyya, K. R. Venugopal, S. S. Iyengar and L. M. Patnaik, MSSS: Most Significant Single-keyword Search over Encrypted Cloud Data, *in Proceedings of the 6th Annual Intrernational Conference on ICT: BigData, Cloud and Security, Singapore*, pp. 43–48, (2015).

[14] S. Raghavendra, C. M. Geeta, Rajkumar Buyya, K. R. Venugopal, S. S. Iyengar and L. M. Patnaik, DRSIG: Domain and Range Specific Index Generation for Encrypted Cloud Data, In *IEEE Proceedings of the International Conference on Computational Techniques in Information and Communication Technologies, ICCTICT 2016*, (2016).

[15] S. Raghavendra, C. M. Geeta, Rajkumar Buyya, K. R. Venugopal, S. S. Iyengar and L. M. Patnaik, MSIGT: Most Significant Index Generation Technique for Cloud Environment, In *IEEE Proceedings of the 12th IEEE India International Conference on* E^3-C^3*(INDICON 2015)*, pp. 1–6, December (2015).

[16] S. Raghavendra, S Girish, C. M. Geeta, Rajkumar Buyya, K. R. Venugopal, S. S. Iyengar and L. M. Patnaik, IGSK: Index Generation on Split Keyword for Search Over Cloud Data, In *IEEE Proceedings of the 2015 International Conference on Computing and Network Communications (CoCoNet'15)*, pp. 380–386, December (2015).

[17] He Tuo and Ma Wenping, An Effective Fuzzy Keyword Search Scheme in Cloud Computing, In *IEEE Proceedings of the 2013 5th International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, pp. 786–789, (2013).

[18] National Science Foundation Research Awards Abstracts 1990–2003, http://kdd.ics.uci.edu/databases/nsfabs/nsfawards.html, (2013).