International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 01 | Jan -2017 www.irjet.net p-ISSN: 2395-0072

Enhanced Security Through Token

Rahil Amin Bhurani[1], Prof. Dr. Girish K. Patnaik[2]

[1]M.E. Student, Department of Computer Engineering, SSBT's COET, Bambhori, Jalgaon(M.S.), India
[2]Professor and Head, Department of Computer Engineering, SSBT's COET, Bambhori, Jalgaon(M.S.), India

---

Abstract - *Access control is a procedure that allows, denies or restricts access to a system. It may, as well, monitor and record all attempts made to access a system. Access Control may also identify users attempting to access a system unauthorized. With Server Computing to meet multiple objectives such as cost, performance, reliability, maintainability, and security, trade-offs have to be made. Any server is vulnerable to an attacker with unlimited time and physical access to the server. Additionally, physical problems could cause the server to have down time. The main goal is to ensure the data integrity and security. Existing Algorithms focus on encrypting data on user side, which incurs high computational and communication costs on user side. To maintain the data securely in distributed environment, Token Generation algorithm in a distributed environment for data files checking is as a secure and dependable Server storage service. A new token generation scheme is suggested to encrypt the user with specified key parameters to make the resource more robust. Token generation scheme will add security for not only authentication but also authorization.*

Key Words: **Man In Middle Attacks, Security, Threats, Authentication, Token Generation Algorithm.**

1. INTRODUCTION

Server computing is the delivery of computing as a service rather than a product. It provides shared resources, software, and information to computers and other devices over a network. This might involve conforming the identity of a person, tracing the origins of an artifact, ensuring that a product is what its packaging and labeling claims to be, or assuring that a computer program is a trusted one [1] [2]. Every person in the world has a trust on web based applications. With the increase in popularity of the Internet the number of frauds and abuses is literally exploding. But the information stored in digital form on web is easily accessible to anyone. Man in Middle attack is a kind of eavesdropper attack [4]. To enhance the security, the introduction of the new additional devices could be costly or the service providers in terms of deployment. Further, there is very little re-use and sharing of additional devices such that the same security token can be used for several systems [3]. To avoid the tedious task of remembering difficult passwords, users often behave less securely by using weak passwords [5].

Server computing provides the storage and supports for outsourcing of data without having the local copy of data or files. Similarly, when a suspicious user tries to access messages in a closed group, our system generates a question from one of the messages posted in that group [8]. In the given Figure 1, the scenario of attacker is shown. The attacker tricks the victim or uses an exploit in order to execute the Switcher. The Switcher then plants the attacker's synchronization token into the Drive. When this first switch is complete, the Switcher copies the original synchronization token into the synced folder. The Drive Application syncs with the attackers account. The attacker then has possession of the victim's synchronization token. The attacker then uses the stolen synchronization token to connect with the victim's file synchronization account. The Switcher tool runs for the second time on the victim's machine, restoring the original synchronization token of the victim, essentially restoring the Drive to its original state. [7]

To maintain the data integrity and data availability researchers proposed several algorithms and methods. The main purpose of token generation algorithm is to ensure the data integrity and security. The suggested scheme of Token generation algorithm which is simple and secure method and less overhead due to few parameters that has to be considered. Challenge verification scheme designed in easy and efficient way to prevent data from Man in middle attacks.

Section 2 describes Literature Survey based on token generation algorithm used in various domains. The solution based on Related Work and analysis of problems in token generation algorithms is given in Section 3. Section 4 gives the conclusion implicating benefits of solution.

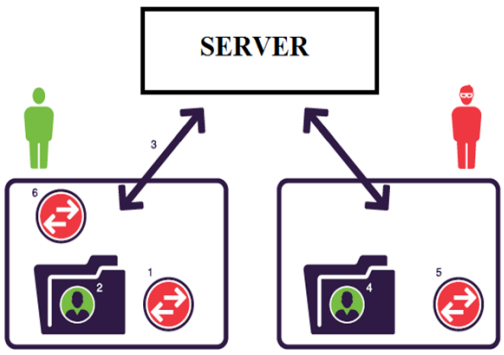

Figure 1: Exiting Attacker Model

2. RELATED WORK

In the past, there was lot of research done on the token generation algorithm and authentication methods. Many schemes have been proposed to enhance authentication and verification, but they may require specialized devices or they may not be always reliable. This Token Generation process is the backbone of the most authentication systems. A disadvantage like compromising of token appears like stealing the synchronization token. Therefore a large requirement to have a strong authentication method is needed to secure the system as possible. This section describes related work of various techniques and methods to secure the data at server side through the use of tokens. To enhance security using token generation algorithm in domains such as Token Generation, Privacy and Secure Data De-Duplication, are described below in related work.

Bharti Dhote and A.M.Kanthe, in [5], proposed a scheme is to build efficient data security model which supports for Data Integrity, to ensure the users data is correct and stored in cloud server, effectively locate the server on which data has been modified by unauthorized user, support for the dynamic data like append, delete, insert, update while retaining the same storage correctness and uses token generation algorithm to pre-computes the verification tokens. This would be a loss of availability. The drawback of this system is upload time and download time of token while pre computation of individual block of server.

Bhagyashree Alhat and Amar Buchade, [19], proposed a concept of token generation and replacing the token by hash value. In cloud storage data is stored in a distributed manner at different servers. Users can remotely store their data in cloud storage without having physical possession of the outsourced data, which encounters security risks in integrity, correctness and availability of the data in the storage system.

P.Srinivas and Rajesh Kumar, in [18], proposed a concept of Secure Data transfer in Cloud Storage Systems using Dynamic Tokens. Though it eliminates the responsibility of local machines to maintain data, there are chances to lose data or it effects from external or internal attacks. Due to Increased bandwidth and less reliable network connections in cloud authors proposed a scheme of Flexible distributed scheme with token generation algorithm for data stored in cloud which is highly efficient against data modification attack and gives Secure and dependable server storage service but no block storage implemented.

Jan Camenisch et al., in [10], proposed a practical scheme to apply the data minimization principle, when the verifiers' authentication logs are subjected to external audits and extended PABC scheme in which verifier further remove attributes from presentation tokens before handing them to an auditor. The Advantage of this scheme is Audited tokens can be linked to the presentation from which they were derived which can be used as a feature when the verifier must be unable to initiate the number of presentations that it performed, and it also is a privacy drawback.

Himika Parmar et al., in [12], proposed an image based authentication scheme on the base of pre-chosen categories of grid of pictures to eliminate the need of text passwords and to generate OTP Token for authorized user after image authentication using Hash Message Authentication Code HMAC based technique. A hash-based OTP starts with the input parameters as synchronization value, username, password, and encrypt them through the cryptographic hash function, which produces the fixed-length password, in the form of OTP. The OTP operates in two modes of delivery text messaging as well as Email services, also has a drawback of spam in the form of email. The Proposed algorithm uses the Time Synchronized OTP Values to generate the token, and has drawback of expiring the token if the session is expired.

Sohil Sharma et al., in [4], proposed a concept of Two Layer Encryption techniques in which the user performs the coarse grained encryption which reduces the costs on user side while the Server provides _ne grained encryption techniques which ensures the confidentiality of the system as well as also reduces costs on user side. The proposed system is to provide user with a safe and secure environment in public clouds to ensure that user receives benefits incurring costs. Using the proposed system the confidentiality of the data will be totally ensured. Hence using the proposed system there is decentralization of attributes and no interference of roles as there was in previous kinds of algorithms. Because encrypting data on user side incurs high computational costs but load increases on server side result in a drawback.

Deepali C. Ghosalkar, in [13], proposed convergent encryption technique to enforce data confidentiality while making De-Duplication feasible because data De-Duplication at server storage reduces the amount of storage space and saves bandwidth. The proposed system encrypts as well as decrypts a data with a convergent key, is obtained by encrypting the hash value of the content of the data. After key generation and data encryption, users retain the keys and send the cipher text to the cloud. The proposed Encryption scheme is deterministic which is derived from data content has a drawback of data modification.

Aparna Ajit Patil and Dhanahree Kulkarni, in [14], proposed a concept of token generation using convergent key for block level data duplication in hybrid cloud systems which uses authorized duplicate check which gives confidentiality of data incurring minimum overhead but with non-web interface. The proposed system encrypts data with a convergent key, the content of the data obtained by computing the cryptographic hash value. After the data encryption and key generation process user retain the keys and send the cipher text to the cloud. The encryption operation is determinative and derived from the data contents, same.

Data copies will generate the same convergent key and hence the same cipher text. A secure proof of ownership protocol is provided to prevent the unauthorized access.

Xin Jin et al., in [15], proposed of Attribute based access control where access requests are evaluated based on the attributes of cloud users and those of objects such as virtual machines, storage volumes, networks, due to lack of flexible model to accommodate diverse policy requirements, the CSP needs a flexible model to accommodate diverse policy requirements, though the scheme proposed is suitable for server model not for administration model.

3. PROPOSED SOLUTION

With reference to literature work, Identification is a process to determine an unknown individual out of many. The proposed solution introduces a new token generation mechanism by using user given attributes and data for authentication to authenticate the user. In proposed solution, token generation methods and mathematical formulation of those methods are described in detail.

3.1 Proposed Approach

An existing system consist drawback of single step verification i.e. less security level is exist in the existing system. The proposed solution overcomes with this drawback by providing more security in the system using two step verification processes. The two step verification process includes token generation with attribute and token generation without attribute methods. The proposed token generation algorithm takes a few parameters in the form of questions and answers and uses attribute base parameters to compute the token. As using the algorithm token is generated at server side, it is difficult to reveal the generation algorithm or scheme. After generating token, token will be ensuring correct answers from authenticated and authorized users. Every user needs to authenticate by providing correct answers. Compromised Tokens retrieved by attackers will also go through the authentication and authorization phase, so the attacks would be nullified with this. The suggested solution of token generation algorithm generates a token by taking the input data from users in the form of question and answers. After taking input from user the token generated in the form of question of which user knows the answer. By providing the correct answer in challenge verification step by algorithm the user is authenticated with server and the starts synchronization with server. Various steps involved for securing client server environment using token generation observed as:

1. Token generation algorithm using question and answers as an input to generate token with additional attribute based policies is a secure method and less overhead due to few parameters that has to be chosen.
2. Challenge verification scheme was designed in easy and efficient way to prevent data from Man in middle attacks and data dependability detection.
3. Servers ensure that the tokens were saved successfully without block modifications. This can be achieved by two way token checking.
4. Token generation algorithm is very cost effective as it does not require any costly certification.
5. It requires very less time consuming process as it perform simple and easy results.
6. It reduces the threat on confidentiality as it stops the disclosure of data from attackers.
7. In case token in stolen, deleted or misplaced by any outsider, will to unable to access the victims account.

3.2 Proposed System Architecture

Architecture is a system that unifies its components into coherent and functional blocks. Figure 2 shows architecture of the proposed system. User requests for token generation at server side by providing the necessary input. Verification is done using the generated token at server side with the user. If token matches then synchronization starts otherwise fails. Initially the user request token to server by providing input data in the form of question and answers. When user accesses the token, the token request the answer followed by the question to the user. Hence the authorized user who has provided input for token generation can give the answer and the synchronization starts. The proposed scheme is highly efficient and resilient against attacks like Man in Middle attack. The Two way verification of token which result in more robust and ensure that data will not be modified before reaching to server. Authentication is the act of establishing something as authentic, i.e. that claims made by or about the subject are true.

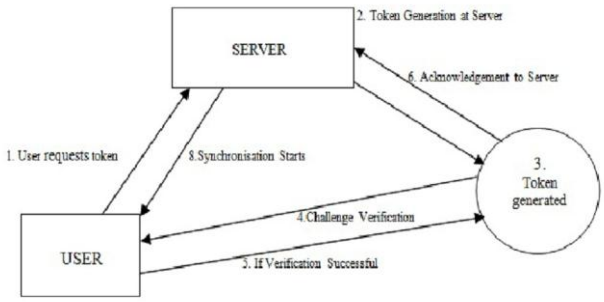

Figure 2: Proposed System Architecture

3.3 Design

The algorithm for the proposed system combines a Encryption based approach with a Token generation algorithm using RC6 Hashing technique [20]. With security reasons rarely server stores user token in plain text form. In the proposed concept, server stores users token in hash form so it cannot be returned in plain text form to unauthorized person. A successful authentication requirements can be interpreted as the result of the calculation hash of the token

sent by the client must be the same with the hash value of token stored at the server side. Hash Function takes data and an initial value as an input and produces the hash value. Cryptography is the ability of keeping message secure form others while sending information between participants. Data Integrity is assuring that data received is same as sent by the sender. Authentication is the ability to assure that communicating party is who that it claims to be. Non-Repudiation is the prevention against the denial by entities involved in the communication. Access Control is the prevention against the unauthorized use of resources. A cryptographic hash function is a mathematical transformation that takes a data of arbitrary length computes a fixed length value also known as hash value, message digest, hash code, hash sum, checksum, etc. "h=H(M)", Where H is Hash Function, M is variable length message; H is fixed size hash value. The hash value is then concatenated with the Question and sends to the user. The user authenticates the token by giving the answer.

3.4 Algorithm

The algorithm for token generation in the proposed system combines encryption based approach using RC6 hashing technique. The two algorithms are proposed. First for token generation without attribute using input data and second for token generation with attribute using input data. The algorithm of both phases is as follows. Ensuring the security and dependability for Server data storage, the aim to design a token generation algorithm for dynamic data verification and operations. To resolve the issues of token generation for Authentication as well as authorization, some schemes have to be introduced for both servers and clients. To achieve data storage correctness and data integrity, token generation takes a few parameters in the form of question and answers given by the authenticated user and uses attribute based parameters and compute the token. Data Synchronization is allowed with token verification. Server is responsible to generate token and stores the token persistently and securely for further verification. The very first challenge of storage correctness is to ensure users that their data are indeed stored appropriately and kept intact all the time at the server is possible with the help of synchronization of token. The second challenge is fast localization of data error which is to effectively locate the malfunctioning server when data corruption has been detected which is possible when the server is compromised and using the challenge verification of token makes a feasible solution. To resolve the issues of token generation for authentication as well as authorization, some schemes have to be introduced for both servers and clients. To achieve data storage correctness and data integrity, token generation takes a few parameters in the form of question and answers given by the authenticated user and uses attribute based parameters and compute the token. Data Synchronization is allowed with token verification. The proposed system provides a solution for these challenges. The input to token generation algorithm is data provided by the user. The algorithm accepts the data and generates the base value for the data. After the base value is generated it is converted into bytes and stored in a temporary token. Then the length of token is calculated that whether it is greater than the input if not then Hash value is generated for the token and added to the temporary token. Again the length of token is checked and final token is generated and stored at server side.

Token Generation Algorithm:

Input: T_b : Base Value for token,
T_h : hash value generated for token,
TR : Random Value,

TR_b : Random Base Value,
TR_h : Random Hash Value,
R_b : Remaining Base Value for parsing,
R_h : Remaining hash value for parsing,
N : Number of token characters,
T_n : Temp Token,
Vr: Total remaining.

Assume, C_n : Count number of digits,
L_c : List of tokens,
TV_c : Total length of token.

1. Input question from user
2. Generate Base value ,
   $T_n = T_b * T_h$;
3. Input answer
   $T_n = TR_b * TR_h$;

DO

4. If question then ,
   $V_r=R_c$;
   $R_b = T_b$;
   $R_h = T_h$;
5. Choose bits of token
6. If $TR_b > R_b$ then skip bits goto step 4
7. Else if $TV_c > Cn$ then skip bits goto step4
8. Else
9. Convert into bits and add to token,
   $L_c = L_c + C_R$
   Add token into previous value
   Add
   $TV_c = TV_c + V_{cr}$
10. Increment the token length
    $T_n = T_n +1$;
11. If $R_b > CR_b$ then generate hash value
    Add token hash value
    $TV_c = TV_c + T_h$
12. Else
13. If $R_h > CR_b$ then Generate hash value for answer and attribute Add token value to previous token
14. $TV_c = TV_c + T_h$
15. Calculate token length
    $T_n = TV_c / C_n$;

Display token generated

16. Decrement bits
    $L_n = L_n-1$

17. Merge the token generated from question and answer
18. Display the final generated token.

While all bits are completed
i.e. while L_n=0

Token Generation With Attribute Algorithm:

Input: T_b : Base Value for token
T_h : hash value generated for token
TR : Random Value
TR_b : Random base value
TR_h : Random Hash Value
R_b : Remaining Base Value for parsing
R_h : Remaining hash value for parsing
N : Number of token characters
T_n : Temp Token

Assume , C_n : count number of digits
L_c : List of tokens
TV_c : Total length of token

1. Input question from user
2. Generate Base value ,
$T_n = T_b * T_h$;
3. Input answer
$T_n = TR_b * TR_h$;
4. Input Attribute
$T_n = R_b * R_h$;

DO

5. If question then ,
$V_r=R_c$;
$R_b = T_b$;
$R_h = T_h$;
6. Choose bits of token
7. If $TR_b > R_b$ then skip bits goto step 4
8. Else if $TV_c > Cn$ then skip bits goto step4
9. Else
10. Convert into bits and add to token,
$L_c = L_c + C_R$

Add token into previous value
$TV_c = TV_c + V_{cr}$

11. Increment the length
$C_n = C_n + 1$;
12. If $R_b > CR_b$ then generate hash value
Add token hash value
$TV_c = TV_c + T_h$
13. Else
14. If $R_h > CR_b$ then

Generate hash value for answer, Add token value to previous token
$TV_c = TV_c + T_r$

15. If $R_h > CR_b$ then

Generate hash value for answer, Add token value to previous token
$TV_c = TV_c + T_r$

16. Calculate token length
$T_n = TV_c / C_n$;

Display token generated

17. Decrement bits
$L_n = L_n - 1$

18. Merge the token generated from question and answer
19. Display the final generated token.

While all bits are completed
i.e. while L_n=0

3.5 RC6 Hashing

RC6 [20] is specified as RC6-w r b where w is the word size in bits, r is the number of rounds and b is the length of the encryption key in bytes. When any other value of w or r is intended in the text, the parameter values will be specified as RC6-w r. Four w-bit registers A, B, C, D contain the initial input plain-text as well as the output ciphertext at the end of encryption. Algorithm presented here has variable parameters i.e. different parameter value can be taken according to the need. Block size (b), Number of rounds (r), length of hash value (l) are the different variable parameters that are the inputs of the algorithm.

RC6 Hashing:

Input: Plaintext stored in four w-bit input registers A; B; C; D
Number r of rounds w-bit round keys S[0; ::: ; 2r + 3]

Output: Ciphertext stored in A; B; C; D

1. B=B+S[0]
2. D=D+S[1]
3. Repeat step 4 to 8 for i=1 to r do
4. t = ((Bx(2B + 1)) <<< $\log_2 w$)
5. u = ((Dx(2D + 1)) <<< $\log_2 w$)
6. A = ((A + t) <<< u) + S[2i]
7. C = ((C + u) <<< t) + S[2i+ 1]
8. (A,B,C,D) = (B,C,D,A)
9. A = A + S[2r + 2]
10. C = C + S[2r + 3]

RC6 De-Hashing:

Input: Ciphertext stored in four w-bit input registers A,B,C,D
Number r of rounds w-bit round keys S[0,...,2r + 3]

Output: Plaintext stored in A,B,C,D

1. C=C-S[2r+3]
2. A=A-S[2r+2]
3. Repeat step 4 to 8 for i=1 to r down to 1 do
4. (A,B,C,D) = (D,A,B,C)
5. u = ((Dx(2D + 1)) <<< $\log_2 w$)
6. t = ((Bx(2B + 1)) <<< $\log_2 w$)
7. C = ((C ⍰ S[2i + 1]) >>> t) + u
8. A = ((A + S[2i]) >>> u) + t
9. D = D - S[1]
10. B = B - S[0]

4. RESULTS AND DISCUSSIONS

Result and discussion section is an primary part of research work. Evaluation of the proposed approach versus existing approach is carried out in the result and discussion section. Result section represents the experimental results of the proposed approach as well as the existing approach. Evaluation of the both the approaches are carried out in the discussion section on the basis of obtained results.

4.1 Experimental Results

Experimental result present the effectiveness of proposed system, in which involvement of authorization and verification is proved better by carrying out experiments. Results are carried out using java. In proposed system considering the drawback of existing system a Man in Middle attack is generated by stealing the security token. The attacker successes to access the token but fails at synchronization step to provide the required answer and Attribute requested by the token. Hence if the attacker possesses the victims token which is stored in a file in terms of hash code encrypted at server side will be unable to decrypt Answer from the token. All the evaluation is done on i3 processor with 4GB RAM and may vary as and when the hardware is changed. The Figure 3 shows the scenario of proposed system where the attacker fails to synchronize the victims account.

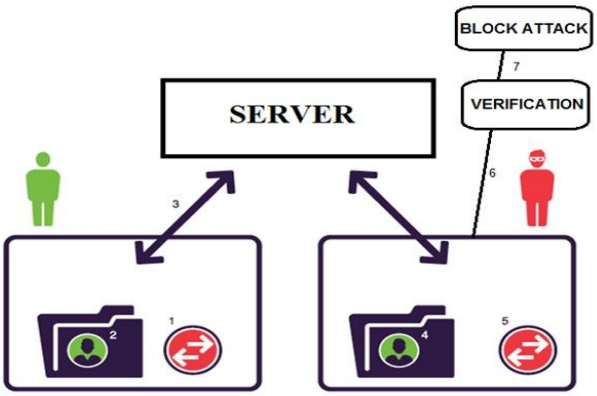

Figure 3: Prevention of MITM Attack

1. The attacker tricks the victim uses an exploit in order to execute the Switcher.
2. The Switcher then plants the attacker's synchronization token into the Drive Application.
3. When this first switch is complete, the Switcher copies the original synchronization token into the synced folder.
4. The Drive Application syncs with the attackers account.
5. The attacker then has possession of the victims synchronization token.
6. The attacker then uses the stolen synchronization token to connect with the victims file synchronization account.
7. Attacker is asked with question and answer generated from token, attacker needs to provide with answer, and verification of answer is done by the token itself.
8. Attacker is blocked for synchronization with stolen token.

4.2 Discussions

The purpose of the discussion section is to state interpretations and opinions, explain the implications of findings in the experimental evaluation. Main role of this section is to clarify the objectives posed in the Introduction, explain how the results support the answers and, how the answers with existing knowledge on the topic. The discussion section is important to know the detail advantage and need about proposed solution. Results of experiments evaluate performance of proposed system at two levels of attacks. Performance is improved as both the attacks Man in the Middle and Authentication attack is prevented. The attacks are prevented by proposed system faced by the attack model having the security token as shown in results section which gives the effectiveness of proposed system.

5. CONCLUSION AND FUTURE WORK

The Proposed system is developed keeping in mind the problem of data security in server storage systems. To ensure the correctness of user's data in storage systems scheme of the token generation using attributes base parameters by the user ensures that the token is generated securely by server. Hence was sent to the server with the purpose of addressing shortcomings of existing authentication solution. The generation of token and storing it to Server ensures confidentiality of data. Data storage on server provides better performance and can easily distribute the data to different server for more data availability. Data Synchronization is easily possible simply by challenge verification of token and utilizing the token with distributed verification achieves the integration of storage correctness insurance and data error localization.
In future, it would be a point of research to increase security by using better user interface or combine these algorithms with more famous algorithms to get better security.

REFERENCES

[1] M. Singhal and S. Tapaswi, "Software tokens based two factor authentication scheme," International Journal of Information and Electronics Engineering, vol. 2, pp. 383-386, 2012.
[2] S. Acharya, A. Polawar, and P.Y.Pawar, "Two factor authentication using smartphone generated one time password," International Journal of Computer Engineering, vol. 11, no. 2, pp. 85-90, 2013.
[3] V. Rao and K. Vedavathi, "Authentication using mobile phone as a security token," International Journal of

Computer Science and Engineering Technology, vol. 1, no. 9, pp. 569-574, 2011.
[4] S. Sharma, V. Peherwar, A. Varma, P. Kode, and T. Bemil, "Preserving security in clouds using ABAC policies," International Journal of Engineering and Science and Computing, vol. 6, no. 4, pp. 3581-3583, 2016.
[5] B. Dhote and A.M.Kanthe, "Secure approach for data in cloud computing," International Journal of Computer Applications, vol. 64, no. 22, pp. 19-24, 2013.
[6] A. Jesudoss and N. Subramaniam, "A survey on authentication attacks and countermeasures in a distributed environment," Indian Journal of Computer Science and Engineering, vol. 5, no. 2, pp. 71-77, 2014.
[7] S. Singh, B. Pandey, R. Srivastava, and N. Rawat, "Cloud computing attacks: A discussion with solutions," Open Journal of Mobile Computing and Cloud Computing, vol. 1, no. 1, pp. 1-4, 2014.
[8] M. Li and K. Tajima, "Automatic generation of authentication questions from private messages," International Conference on Web Intelligence and Intelligent Agent Technology, vol. 10, pp. 505-510, 2015.
[9] A. Jayarana, A. Cahyawan, and G. Sasmita, "Dynamic mobile token for web security using md5 and one time password method," International Journal of Computer Applications, vol. 55, no. 6, pp. 1-6, 2012.
[10] J. Camenisch, A. Lehmann, G. Neven, and A. Rial, "Privacy-preserving auditing for attribute-based credentials," IBM Research, pp. 11-30, 2011.
[11] T. Graf.F and Prema.P, "Secure collaborative privacy in cloud data with advanced symmetric key block algorithm," International Journal of Computer Science and Engineering, vol. 2, no. 22, pp. 45-249, 2015.
[12] H. Parmar, N. Nainan, and S. Thaseen, "Generation of secure one-time password based on image authentication," International Journal of Computer Science and Information Technology, vol. 2, pp. 195-206, 2012.
[13] D. C. Ghosalkar, "Implementation idea for secure data de duplication using hybrid cloud approach," International Journal of Computer Science Trends and Technology, vol. 4, no. 1, pp. 136-139, 2016.
[14] A. A. Patil and D. Kulkarni, "Block level data duplication on hybrid cloud storage system," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, no. 8, pp. 340-345, 2015.
[15] X. Jin, Krishnan, and R. Sandhu, "Role and attribute based collaborative administration of intra-tenant cloud iaas," International Journal of Computer Networks, vol. 3, no. 3, pp. 159-165, 2011.
[16] S.B.Patil, Y. A. Kumbhar, and S. Mane, "Data deduplication using hybrid cloud," International Research Journal of Engineering and Technology, vol. 2, no. 7, pp. 1009-1012, 2015.
[17] M. Ghodke, P. Bais, A. Saha, P. Singh, and G.M.Gaikwad, "Securely eradicating duplication by generating _le tags and tokens over hybrid cloud using security algorithm," International Journal of Advanced Research in Computer Communications Engineering, vol. 5, no. 3, pp. 151-153, 2016.
[18] P. Srinivas and R. Kumar, "Secure data transfer in cloud storage systems using dynamic tokens," International Journal of Research in Computer and Communication Technology, vol. 2, no. 1, pp. 6-10, 2013.
[19] B. Alhat and A. Buchade, "Revisiting secure cloud storage by replacing token generation with SHA," International Journal of Advance Foundation And Research In Science Engineering, vol. 1, no. 12, pp. 14-20, 2015.
[20] K. Aggarwal and H. K. Verma, "Variable length hash algorithm using rc6," International Conference on Advances in Computer Engineering and Applications, vol. 50, no. 4, pp. 450-456, 2015.

BIOGRAPHIES

Rahil Amin Bhurani is a Student pursuing M.E (Computer Science and Engg) Final Semester in Department of Computer Engineering, SSBT's COET, Bambhori, Jalgaon (M.S.), India.

Prof. Dr. Girish K. Patnaik is a Professor and Head in Department of Computer Engineering, SSBT's COET, Bambhori, Jalgaon(M.S.),India.