Imperial Journal of Interdisciplinary Research (IJIR)
Vol.2, Issue-1 , 2016
ISSN : 2454-1362 , www.onlinejournal.in

Protecting Data in Cloud Storage Using Blowfish Encryption Algorithm and Image-Based One-Time Password

M Rama Raju
Asst. Professor
CSE Department,
Christu Jyothi Institute of Technology & Science.

J Purna Prakash
Asst. Professor
CSE Department,
Christu Jyothi Institute of Technology & Science.

Abstract
One of the primary usage of cloud computing is data storage. Cloud provides huge capacity of storage for cloud user'. To store and retrieve their data at anytime or anywhere it should be reliable and flexible. Currently many enterprises have started using cloud storage due to its advantages. But the problem lie in data security, data privacy and other data protection issues. It is a major setback for security and privacy of data storage in the field of cloud computing. This paper proposes an encryption algorithm to address the security and privacy issue in cloud storage in order to protect the data stored in the cloud.

Keywords: *Cloud Storage, Security, Privacy, Encryption Algorithm.*

1. Introduction

Cloud is basically the collection of computers on the internet that companies are using to offer their services. One cloud service that is being offered is a revolutionary storage method for your data. From music files to pictures to sensitive documents, the cloud invisibly backs up your files and folders and alleviates the potentially endless and costly search for extra storage space. An alternative to buying an external hard drive or deleting old files to make room for new ones, cloud storage is convenient and cost-effective. It works by storing your files on a server out in the internet somewhere rather than on your local hard drive. This allows us to back up, sync, and access our data across multiple devices as long as they have internet capability.

However, if you wish to store information virtually, you must consider the added risk that your information may be accessible to other—potentially people who you do not wish to have access. Below, we outline a few security risks to take into account and how to protect yourself and your data.

Cloud computing [4] is a relatively new tool for the average consumer. It is important to explore the service that most fits your needs. A few popular options when deciding which company to use:

- Dropbox
- SugarSync
- Amazon Cloud Drive
- Windows Live mMesh
- Box.net
- SpiderOak

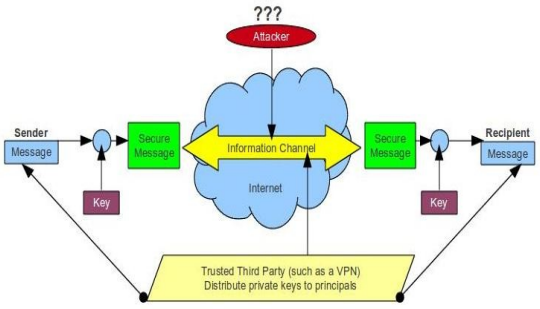

Fig1: Symmetric Encryption Technique

Information is costing the world's economies billions of dollars each year while the great majority of the teaching and research information is and must remain public many of us also routinely deal with information that is sensitive and that we are required to protect sensitive information with which they may come into contact protection of information. Information is held electronically and might be found on our computers laptops tablets and Smartphone's data protection standards to help our community understand and meet these requirements is to develop the data protection standards they are a simplified setup requirements designed to guide and help insure information security compliance. In fig1, shows symmetric encryption technique or single key encryption which provides trusted third party that distribute private keys to clients. Same key is used for encryption and decryption.

.

2. Cloud Risks

Shared access

One of the key tenets of public cloud computing [6] is multitenancy, meaning that multiple, usually unrelated customers share the same computing resources: CPU, storage, memory, namespace, and physical building.

Multitenancy is a huge known unknown for most of us. It's not just the risk of our private data accidentally leaking to other tenants, but the additional risks of sharing resources. Multitenancy exploits are very worrisome because one flaw could allow another tenant or attacker to see all other data or to assume the identity of other clients.

Several new classes of vulnerabilities derive from the shared nature of the cloud. Researchers have been able to recover other tenants' data from what was supposed to be new storage space. Other researchers have been able to peek into other tenants' memory and IP address space. A few have been able to take over another tenant's computing resources in totality by simply predicting what IP or MAC addresses were assigned.

Multitenancy security issues are just now becoming important to most of us, and the vulnerabilities within are starting to be explored. The best precursor example is a single website placed on a Web server with hundreds or even thousands of other, unrelated websites. If history is any guide -- it usually is -- multitenancy will be a big problem over the long haul.

Virtual exploits

Every large cloud provider is a huge user of virtualization. However, it holds every risk posed by physical machines, plus its own unique threats, including exploits that target the virtual server hosts and the guests. You have four main types of virtual exploit risks: server host only, guest to guest, host to guest, and guest to host. All of them are largely unknown and uncalculated in most people's risk models.

When I talk to senior management about virtual risk issues, their eyes glaze over. Many have said to me that the risks are overblown or exploits are unheard of. I usually tell them to check out their own virtualization software vendor's patch list. It isn't pretty.

To up the ante, the cloud customer typically has no idea what virtualization products or management tools the vendor is running. To shed some light on this risk, ask your vendor the following questions: What virtualization software do you run? What version is it on now? Who patches the virtualization host and how often? Who can log into each virtualization host and guest?

Authentication, authorization, and access control

Obviously, your cloud vendor's choice of authentication, authorization, and access control mechanisms is crucial, but a lot depends on process as well. How often do they look for and remove stale accounts? How many privileged accounts can access their systems -- and your data? What type of authentication is required by privileged users? Does your company share a common namespace with the vendor and/or indirectly with other tenants? Shared namespaces and authentication to create single-sign-on (SSO) experiences are great for productivity, but substantially increase risk.

Data protection is another huge concern. If data encryption is used and enforced, are private keys shared among tenants? Who and how many people on the cloud vendor's team can see your data? Where your data is physically stored? How is it handled when no longer needed? I'm not sure how many cloud vendors would be willing to share detailed answers to these questions, but you have to at least ask if you want to find out what is known and unknown.

Availability

When you're a customer of a public cloud provider, redundancy and fault tolerance are not under your control. Heck, usually what's provided and how it's done are not disclosed. It's completely opaque. Every cloud service claims to have fantastic fault tolerance and availability, yet month after month we see the biggest and the best go down for hours or even days with service interruptions.

Of even bigger concern are the few instances in which customers have lost data, either due to an issue with the cloud provider or with malicious attackers. The cloud vendor usually states that they do awesome, triple-protected data backups. But even in cases where vendors said that data backups were guaranteed, they've lost data -- permanently. If possible, your company should always back up the data it's sharing with the cloud or at least insist on legalese that has the right amount of damages built in if that data is lost forever.

Ownership

This risk comes as a surprise to many cloud customers, but often the customer is not the only owner of the data. Many public cloud providers, including the largest and best known, have clauses in their contracts that explicitly states that the data stored is the provider's -- not the customer's.

A cloud vendor like owning the data because it gives them more legal protection if something goes wrong. Plus, they can search and mine customer data to create additional revenue opportunities for themselves. I've even read of a few cases where a cloud vendor went out of business, then sold their customers' private data as part of their assets to the next buyer. It's shocking. Make sure you have this known unknown on lockdown: Who owns your data and what can the cloud provider do with it?

3. Symmetric Encryption

Blowfish [7] is a symmetric block encryption algorithm designed in consideration with,

- **Fast:** It encrypts data on large 32-bit microprocessors at a rate of 26 clock cycles per byte.
- **Compact:** It can run in less than 5K of memory.
- **Simple:** It uses addition, XOR, lookup table with 32-bit operands.
- **Secure:** The key length is variable, it can be in the range of 32~448 bits: default 128 bits key length.

It is suitable for applications where the key does not change often, like communication link or an automatic file encryptor.

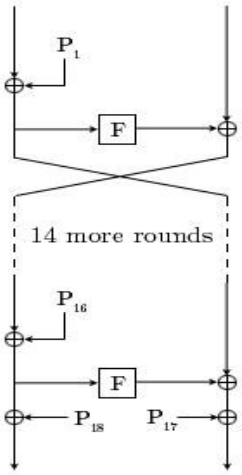

Fig2: The Feistel structure of Blowfish

3.1 Description of Algorithm:

Blowfish symmetric block cipher algorithm encrypts block data of 64-bits at a time. It will follows the feistel network (in fig2,) and this algorithm is divided into two parts.

1. Key-expansion
2. Data Encryption

3.1.1 Key-expansion:

Blowfish uses a large number of subkeys. These keys must be precomputed before any data encryption or decryption.

The P-array consists of 18 32-bit subkeys:

P1, P2,..., P18.

There are four 32-bit S-boxes with 256 entries each:

S1,0, S1,1,..., S1,255;
S2,0, S2,1,..,, S2,255;
S3,0, S3,1,..., S3,255;
S4,0, S4,1,..,, S4,255.

Generating the Subkeys:

The subkeys are calculated using the Blowfish algorithm:

1. Initialize first the P-array and then the four S-boxes, in order, with a fixed string. This string consists of the hexadecimal digits of pi (less the initial 3): P1 = 0x243f6a88, P2 = 0x85a308d3, P3 = 0x13198a2e, P4 = 0x03707344, etc.
2. XOR P1 with the first 32 bits of the key, XOR P2 with the second 32-bits of the key, and so on for all bits of the key (possibly up to P14). Repeatedly cycle through the key bits until the entire P-array has been XORed with key bits. (For every short key, there is at least one equivalent longer key; for example, if A is a 64-bit key, then AA, AAA, etc., are equivalent keys.)
3. Encrypt the all-zero string with the Blowfish algorithm, using the subkeys described in steps (1) and (2).
4. Replace P1 and P2 with the output of step (3).
5. Encrypt the output of step (3) using the Blowfish algorithm with the modified subkeys.
6. Replace P3 and P4 with the output of step (5).
7. Continue the process, replacing all entries of the P array, and then all four S-boxes in order, with the output of the continuously changing Blowfish algorithm.

In total, 521 iterations are required to generate all required subkeys. Applications can store the subkeys rather than execute this derivation process multiple times.

3.1.2 Data Encryption:

It is having a function to iterate 16 times of network. Each round consists of key-dependent permutation and a key and data-dependent substitution. All operations are

XORs and additions on 32-bit words. The only additional operations are four indexed array data lookup tables for each round.

```
----------------------------------------
 Algorithm:Blowfish Encryption
----------------------------------------

Divide x into two 32-bit halves: xL, xR
For i = 1to 16:
   xL = XL XOR Pi
   xR = F(XL) XOR xR
  Swap XL and xR
  Swap XL and xR (Undo the last swap.)
   xR = xR XOR P17
   xL = xL XOR P18
 Recombine xL and xR
```

4. Proposed scheme

The proposed solution blowfish algorithm involves image based authentication with OTP generation method.

4.1 Image Based Authentication

The Image-based authentication [3] is based on Recognition Techniques. When the user registers for first time in a web site they select set of images that are easy to remember, such as natural scenery, automobiles etc. Every time the user logs into the site, they are provided with a grid of images that is randomly generated. The user can identify the images that were previously selected by him. It is significantly easier for the user because they need to remember a few simple images only.IBA is based on a user's successful identification of his set of images. When the user logins for the first time, the website displays a grid of images, which consists of images from the user's password set mixed with other images. The user is authenticated by correctly identifying the password images. Performing brute force attacks or other attacks on such systems is very difficult. A set of different images are selected to authenticate the user. The Image Identification Set (IIS), for each user is then stored at the Authentication System. When a user logins, the IIS for that user is retrieved and used to authenticate that particular user. The system does not store the images but the category of the images are stored in IIS as images are large files. This technique is also more secure and requires less memory. If this step is successful, next OTP [1] is generated and send to the user email-id or registered mobile number.

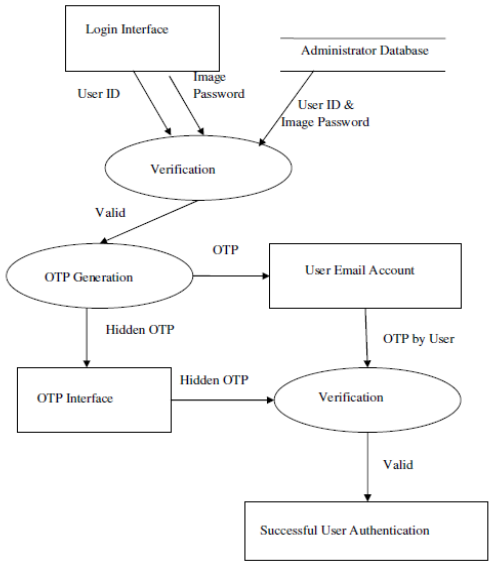

Fig 3: Authentication using OTP

New Approach:
We explain our new approach as:
1. When user login servers take its Email ID and Password and authenticate the user.

2. Server simply encrypts the Id and password and gives that output to OTP generator.
3. OTP generator starts its work.OTP selects two alphabets from encrypted data and use blowfish algorithm.
The function F is as follows:
For XL, into four 8-bit: a, b, c and d.
F (XL) = ((S1, a + S2, b mod 232) XOR S3, c) + S4, c mod 232
Subkeys are calculated using the Blowfish algorithm is as follows:
1. First initialitiation P-array and then the four S-boxes in sequence with a fixed string. This string consists of hexadecimal digits of Pi.
2. XOR P1 with the first 32-bit key, XOR P2 with the second 32-bit key, and so for each bit of the key (to P18). Repeat the key bits until the entire P-array has been XORed with key bits.
3. Encrypt the all-zero string with the Blowfish algorithm, using the subkeys described in steps (1) and (2).
4. Replace P1 and P2 with the output of step (3).
5. Encrypt the output of step (3) using the Blowfish algorithm with the modified subkeys.
6. Replace P3 and P4 with the output of step (5).
7. Continue the process, replacing all elements of the P-array, then all four S-boxes in order, with the output changing kontiyu Blowfish algorithm.
Finally we have now newly generated ID of encrypted id and password in database. Next time when user login then that ID is given to OTP generator for generating password.

5. Conclusion

One time password is an efficient technique that generate random password each time for users. If user lost their pervious password then there is no need of worry for them because OTP give them new password for each session.OTP prevent user id from replay or eavesdropping attack. Earlier OTP is generated using HMAC, One way hash function and Ping Pong stream cipher , in which input is given to OTP generator as challenge and it generate random password. In our work we propose a method of generating OTP generator using blowfish algorithm. In future more work should be done on how to provide more security in this approach.

REFERENCE

[1] Balkis Hamdane, Ahmed Serhrouchni, Adrien Montfaucon, Sihem Guemara.” Using the HMAC-Based One-Time Password Algorithm for TLS Authentication” 978-1-4577-0737-7/11/ ©2011 IEEE
[2] D. M’Raihi, M. Bellare, F. Hoornaert, and D. Naccache, “HOTP: An HMAC based one-time password algorithm, RFC 4226”, Dec. 2005
[3] S. Akula and V. Devisetty, "Image Based Registration and Authentication System," in Proceedings of Midwest Instruction and Computing Symposium, 2004. Karen Scarfone, Murugiah Souppaya, Paul Hoffman, “Guide to Security for Full Virtualization Technologies “,http://csrc.nist.gov/publications/nistpubs/800-125/SP800-125-final.pdf , NIST, 2011
[4] Eman M.Mohamed, Hatem S.Abdelkader and Sherif El-Etriby,“Data Security Model for Cloud Computing”, The Twelfth
International Conference on Networks, ISBN: 978-1-61208-245-5,pp 66-74, 2013.
[5] Peter Mell, Tim Grance, “Effectively and Securely Using the Cloud Computing Paradigm”, NIST, Information Technology
Laboratory,http://www.csrc.nist.gov/groups/SNS/cloud-comput ing/cloudcomputing- v26.ppt. 2009.
[6] Pankaj Arora et al., ”Cloud Computing Security Issues in Infrastructure as a Service”, International Journal of Advanced
Research in Computer Science and Software Engineering, Volume 2, Issue 1, 2012.
[7] B. Schneier, Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish) *Fast Software Encryption, Cambridge Security Workshop Proceedings(December 1993)*, Springer-Verlag, 1994, pp. 191-204.