ISSN:0975-9646

Dhanashri Bamane et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (6) , 2015, 5090-5092

Implementation of Efficient Provable Data Possession

Dhanashri Bamane[#1], Prof. Vinayak Pottigar[*2], Prof. Subhash Pingale[#3]

[#]*Computer Science And Engineering Department, Solapur University,Solapur,India*
SKN Sinhgad College of Engineering,Korti,Pandharpur,India

***Abstract*— Provable data possession (PDP) is a technique for ensuring the integrity of data in storage outsourcing. Storage outsourcing is arising trend which prompts a Number of interesting security issues, many of which have been extensively investigated in the past. The PDP word recently added in research literature. The main issue is how to frequently, efficiently and securely verify that a storage server is faithfully storing its client's (potentially very large) outsourced data. The storage Server is assumed to be un-trusted in terns of both security and reliability.**

In this paper, we address the construction and implementation of an efficient PDP scheme for cloud storage to support the scalability of service and data migration in which we not require bulk encryption.

Keywords — PDP; Cloud; Dataoutsourcing;Ecnryptioin & Decryption;

I. INTRODUCTION

Now a day's data outsourcing has become very popular. In General the meaning of the data outsourcing is data owner (Client) moves the data to third party (Server) which is supposed to charge fee for keeping the data. The server is faithful for storing and allowing retrieving data based on the owner demand. The purpose of data to move from local machine to server is reduction of cost money as well as memory. The Servers provides scalable storing space where owner can store more data. The responsibility of Third party is to retain and make available if the demand comes which also includes maintenance. The number of research has made to increase the security while outsourcing the data. How to efficiently and securely ensure that the server returns correct and complete results in response to its clients' queries [1, 2]. Later research focused on outsourcing encrypted data (placing even less trust In the server) and associated difficult problems mainly having to do with efficient querying over encrypted domain [3,4,5,6]. More recently, however, the problem of Provable Data Possession (PDP)– is also some times referred to as Proof of Data retrieve ability (POR)–has popped up in the research literature. The central goal in PDP is to allow a client to efficiently, frequently and securely verify that a server– who purportedly stores client's potentially very large amount of data – is not cheating the client. In this context, cheating means that the server Might delete some of the data or it might not store all Data in fast storage, e.g., place it on CDs or other tertiary off – line media. It is important to note that a storage server might not be malicious; instead, it might be simply un-reliable and lose or in advertently corrupt hosted data. An effective PDP technique must be equally applicable to malicious and unreliable servers. The problem is further complicated by the fact that the client might be a small device (e.g., a PDA or a cell-phone) with limited CPU, battery power and communication facilities. Hence, the need to minimize bandwidth and local computation overhead for the client in performing each verification. The first [7] is a public-key-based technique allowing any verifier (not just the client) to query the server and obtain an interactive proof of data possession. This property is called public verifiability. The POR scheme [8] uses special blocks (called sentinels) hidden among other blocks in the data.

II. MOTIVATION

The maintaining huge amount of data within organization is big challenge even if maintain, the large amount of hardware must continuously update. To maintain hardware is not the optimal solution compare to outsourcing of data [9] . During outsourcing of data security measure should consider. The PDP is Distributed data store system. A potentially interesting new angle for motivating Secure and efficient PDP techniques is the outsourcing of personal digital content.

III. CONTRIBUTION

This paper contributes in Efficiency and Security. The proposed scheme is based on Rijndael symmetric key encryption/Decryption. The PDP is requiring symmetric key for both phases i.e. setup phase and verification phase. However, our scheme is more efficient than POR as it requires no bulk encryption of out sourced data and no data expansion due to additional sentinel blocks.

IV. ROADMAP

The next section introduces our approach to provable data possession and discusses its effectiveness and security;

V. PROPOSED PDP SCHEME

The proposed system is consisting of two phases' namely setup phase and verification phase.

A. Notation

- D: Outsourced data, represent single contiguous file d; With equal block D[1], D[2], D[3]….D[d]
- OWN : Owner of Data(Client)
- SRV : Server
- H(.): Cryptographic hash function i.e. md5.

- $AE_{key}(.)$: Symmetric Encryption Key(Rijndael)
- $AE^{-1}_{key(.)}$:Decryption Function
- $f_{key}(.)$: pseudo-random Function
- $g_{key}(.)$: pseudo-random permutation

B. Setup Phase

D is divided into d blocks.
We need time t to store the d blocks on the server.

Algorithm **SetupPhase** (ContiguousFileD)
{
Choose Parameter c,l,k,L and function g,f
Choose the number t of tokens;
Choose the number r of indices per verification ;
Generate randomly master keys
W;Z;K Є $\{0,1\}^k$.
for i:=1 to t step 1
{
Generate $k_i = f_W(i)$ and $c_i = f_Z(i)$
Compute vi = H(ci ;
$D[gk_i(1)],\ldots..D[gk_i(r)]$)
Compute $v'_I = AE_K(i , vi)$
}
Sendto SRV : (D,{[I,v'] for 1≤ i≤ t});
}
C = log t, l= log d, d = number of blocks. L= cipher block, g = compute permutation for indices.
f used to compute g.
W,Z,K are the master secret key. [10]

C. Implementation

File Uploading with Symmetric algorithm
//C# Code snippets for uploading the data the size supports upto 2 GB File

```
RijndaelManaged myAlg = new RijndaelManaged();
byte[] salt = Encoding.ASCII.GetBytes("this is symmetric encryption algorithm");
Rfc2898DeriveBytes key = new Rfc2898DeriveBytes(wbTxtPassword.Text, salt);
byte[] Key = key.GetBytes(myAlg.KeySize / 8);
byte[] IV = key.GetBytes(myAlg.BlockSize / 8);
byte[] inputfile = Misc.ReadFully(wbFileUpload.PostedFile.InputStream);
intnblock = inputFile/256;
for(int i=0; i<255;i++)
        byte[256] block = inputfile[i];
byte[] encryptedfile =Misc.Encrypt(block);
for(i=256;i<inputfile.length;i++)
        encryptedFile[i] = inputfile[i];
blobBlock.UploadFromByteArray(encryptedfile, 0, encryptedfile.Length);
```

D. Verify Phase

Algorithm Verificationphase(Challenge I)
{
OWN computes ki = $f_W(i)$ and ci = $f_Z(i)$
OWN sends { ki, ci} to **SRV**
SRV computes
z = H(ci, D[gki(1)] ,……,D[gki(r)])
SRV sends {z; v'_i}to**OWN**
OWN extracts v fromv'_i . If decryption fails or v≠ (I,z) then reject
}

E. Implementation

//Code snippents for downloading the data

```
Byte[] rowByte = new Byte[blobBlock.Properties.Length];
RijndaelManagedmyAlg = new RijndaelManaged();
byte[] salt = Encoding.ASCII.GetBytes("this is symmetric encryption algorithm");
Rfc2898DeriveBytes key = new Rfc2898DeriveBytes(txtEncryptionPassDown.Text, salt);
byte[] Key = key.GetBytes(myAlg.KeySize / 8);
byte[] IV = key.GetBytes(myAlg.BlockSize / 8);
Misc.Key = Key;
Misc.IV = IV;
MemoryStreamms = new MemoryStream();
blobBlock.DownloadToStream(ms);
rowByte = Misc.Decrypt(ms.ToArray());
if (rowByte != null)
    {
Response.Clear();
Response.ClearHeaders();
Response.ClearContent();
Response.ContentType = blobBlock.Properties.ContentType;
        Response.AppendHeader("Content-Disposition", "attachment;filename=" + Misc.FileDownload);
Response.AddHeader("content-length", rowByte.Length.ToString());
Response.BinaryWrite(rowByte);
Context.ApplicationInstance.CompleteRequest();
    }
```

VI. CONCLUSION

We developed and presented a step-by-step design of A very light-weight and provably secure PDP scheme. It Surpasses prior work on several counts, including storage, bandwidth and computation overheads as well as the support for dynamic operations. However, since it is based upon symmetric key cryptography, it is unsuitable for public (third-party) verification. A natural solution to this would be a hybrid scheme combining elements of [7] and our scheme. To summarize, the work described in this paper represents an important step forward towards practical PDP techniques. We expect that the salient features of our scheme (very low cost and support for dynamic outsourced data) make it attractive for realistic applications.

ACKNOWLEDGMENT

I would like to thank my guide Prof. Vinayak Pottigar and co-guide Prof. Subhash Pingle for their valuable contribution in completing my work. I would also express thank to my family for moral support.

REFERENCE

[1]. P.Devanbu,M.Gertz,C.Martel,andS.Stubblebine,"Authentic third-party data publication,"in IFIPDB-Sec'03, also in Journal of Computer Security,Vol.11,No.3,pages291-314,2003,2003.

[2]. E.Mykletun, M.Narasimha, and G.Tsudik, "Authentication and integrity in outsourced databases," in ISOCNDSS'04,2004.

[3]. H.Hacig¨ um¨ us, B.Iyer, C.Li, and S.Mehrotra, "Executting sql over encrypted data in the database- service- Provider model," in ACMSIGMOD'02,2002.

[4]. D.Song, D.Wagner, and A.Perrig, "Practical techniques for search eson encrypted data," in IEEES&P'00,2000.

[5]. D.Boneh, G.diCrescenzo, R.Ostrovsky, and G.Persiano, "Public key encryption with key-word search," in EUROCRYPT'04,2004.

[6]. P.Golle, J.Staddon, and B.Waters, "Secure conjunctive keyword search over encrypted data," in ACNS'04,2004.

[7]. G.Ateniese, R.Burns, R.Curtmola, J.Herring, L.Kissner, Z.Peterson, and D.Song, "Provable data possession Atun trusted stores," in ACMCCS'07

[8]. A.Juelsand B.Kaliski, "PORs: Proofs of retrieve ability For large files,"in ACMCCS'07

[9]. JohnF.Gantz, David Reinsel, Christopher Chute, Wolf-Gang Schlichting, John McArthur, Stephen Minton, IridaXheneti, AnnaToncheva, and Alex Manfrediz, "The expanding digital universe: A forecast of world wide information growth through 2010. IDC white paper—sponsored by EMC," Tech.Rep., March2007, http://www.emc.com/about/destination/digital_universe/.

[10]. Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W.Hong, "Freenet:a distributed anonymous information storage and retrieval system," in International workshop on Designing privacy enhancing technologies,NewYork,NY,USA,2001,pp.46–66,Springer-VerlagNewYork,Inc.