VOL. 10, NO. 14, AUGUST 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences
©2006-2015 Asian Research Publishing Network (ARPN). All rights reserved.
www.arpnjournals.com

SECURED DATA STORAGE DESIGN USING CRYPTOGRAPHY

V. Jayanthi and Maria Anu
Sathyabama University, Chennai, India
E-Mail: balkiparu@gmail.com

ABSTRACT

Cryptography is the practice and study of techniques for secure communication in the presence of third parties (called adversaries). More generally, it is about constructing and analyzing protocols that overcome the influence of adversaries and that are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation. Applications of cryptography include ATM cards, computer passwords, and electronic commerce. In this paper, we implement the self-generating key approach it is proposed to use an algorithm stored in a procedure in the SYS schema which will be made executable by PUBLIC with a public synonym. However only the DBA can view or edit the procedure thus preserving the secrecy. The algorithm will generate a RAW key value using the filename of the Binary Large OBject (BLOB) or CLOB and the date of uploading into the LOB with a shuffling algorithm stored in the procedure. This is used as the encryption key.

Keywords: Cryptography, BLOB, CLOB, symmetric, asymmetric.

1. INTRODUCTION

Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key (or, less commonly, in which their keys are different, but related in an easily computable way). This was the only kind of encryption publicly known until June 2012. Symmetric key ciphers are implemented as either block ciphers or stream ciphers. A block cipher enciphers input in blocks of plaintext as opposed to individual characters, the input form used by a stream cipher. The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are block cipher designs which have been designated cryptography standards by the US government (though DES's designation was finally withdrawn after the AES was adopted). Despite its deprecation as an official standard, DES (especially its still-approved and much more secure triple-DES variant) remains quite popular; it is used across a wide range of applications, from ATM encryption to e-mail privacy and secure remote access. Many other block ciphers have been designed and released, with considerable variation in quality. Many have been thoroughly broken, such as FEAL. In Oracle 10g the package DBMS_CRYPTO provides an interface to encrypt and decrypt stored data, and can be used in conjunction with PL/SQL programs running network communications. It provides support for several industry-standard encryption and hashing algorithms, including the Advanced Encryption Standard (AES) encryption algorithm.

Earlier Cryptography was meant encryption, the conversion of information from a readable state to apparent nonsense. The originator of an encrypted message shared the decoding technique needed to recover the original information only with intended recipients, thereby precluding unwanted persons to do the same. Since the computer-era, the methods used to carry out cryptology have become increasingly complex and its application more widespread. Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system, but it is infeasible to do so by any known practical means. These schemes are therefore termed computationally secure; theoretical advances, e.g., improvements in integer factorization algorithms, and faster computing technology require these solutions to be continually adapted.

In this paper, we propose a concept to prevent unauthorized users from accessing the sensitive data as it is subject to unauthorized disclose and access after being outsourced, an end-to end security is provided by cryptographic protocols which are executed by the file owner to prevent proxy servers and unauthorized users from modifying and accessing the sensitive files and also identity-based secure distributed data storage schemes are a special kind of distributed data storage schemes where users are identified by their identities and can communicate without the need of verifying the public key certificates.

2. LITERATURE REVIEW

L. Bouganim and P. Pucheral propose a database servers arouse user's suspicion because no one can fully trust traditional security mechanisms against more and more frequent. Malicious attacks and no one can be fully confident on an invisible DBA administering confidential data. Software acts as an incorruptible mediator between a client and a server hosting an encrypted database. U. Maheshwari, R. Vingralek, and W. Shapiro propose a concept that is the database is encrypted and validated

©2006-2015 Asian Research Publishing Network (ARPN). All rights reserved.

against a collision-resistant hash kept in trusted storage, so untrusted programs cannot read the database or modify it undetectably. The implementation exploits synergies between hashing and log-structured storage. The model is powerful enough to support higher-level database functions such as transactions, backups, and indexing.

G. Ateniese, K. Fu, M. Green, and S. Hohenberger implement a new re-encryption schemes that realize a stronger notion of security, and we demonstrate the usefulness of proxy re-encryption as a method of adding access control to a secure file system. Performance measurements of their experimental file system demonstrate that proxy re-encryption can work effectively. S.D.C. di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, and P.Samarati introduce a shuffle index structure, which adapts traditional B+-trees. It exhibits a limited performance cost, thus resulting effectively usable in practice. Data owner outsources data to an external honest-but-curious server, and accesses their data by submitting requests to a client that directly interacts with the server. Juels and B.S. Kalki Jr propose a concept to formally prove the security of an (optimized) variant of the bounded-use scheme without making any simplifying assumptions on the behavior of the adversary, Build the first bounded-use scheme with information-theoretic security. It remotely stored data is rarely accessed and also interested in schemes with significantly better efficiency.

3. PROPOSED SYSTEM ARCHITECTURE

The following figure shows the architectural diagram of proposed concept.

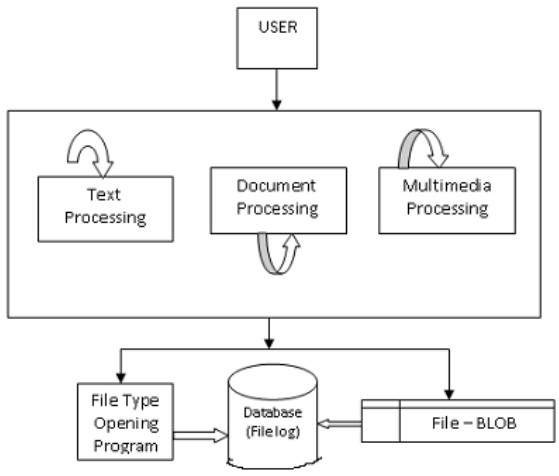

Figure-1. System architecture.

Initially user form displays the User Master information in the HR database and can be used to Add / Modify / Delete / Query User Master Records. The Cryptography Process Menu form has 4 options - Manage Text LOBs which invokes the Text LOB Processing Form, Manage Binary LOBs which invokes the Menu footer the Document LOBs Processing form, Manage Multimedia LOBs which invokes the Multimedia LOBs Processing Form and the Exit Menu Option. Clicking Exit at the top toolbar or the exit button exits the menu to the Cryptography Main Menu Form. Users who login correctly are directly brought to this menu instead of Cryptography Main Menu.

Text Lobs Processing form is used to upload OS Text Files into CLOBs and then encrypt these to BLOBs and store in the DBMS and also to decrypt BLOBs in the DBMS and write them to the Long Text field File Text for viewing. In order to upload text files to the DBMS the user enters the File Name the Key mode (shown in the next screen) and the encryption algorithm (in the next screen). If User-specified key mode is selected the user has to enter the key in the Key Value field (in the next screen) which has to be minimum 8 characters for DES and 32 characters for AES. Then on clicking the Upload File button the file is uploaded and encrypted to a BLOB in the DB Table.

Next, Document Lobs Processing form is used to upload OS PDF and MS-Office Files into BLOBs and then encrypt these to BLOBs and store in the DBMS and also to decrypt BLOBs in the DBMS and write them to the OS Files and invoke the respective opening program for viewing. In order to upload text files to the DBMS the user enters the File Name the Key mode (shown in the next screen) and the encryption algorithm (in the next screen). If User-specified key mode is selected the user has to enter the key in the Key Value field (in the next screen) which has to be minimum 8 characters for DES and 32 characters for AES. Then on clicking the Upload File button the file is uploaded and encrypted to a BLOB in the DB Table.

Finally, Multimedia Lobs Processing form is used to upload OS JPG and audio/video Files into BLOBs and then encrypt these to BLOBs and store in the DBMS and also to decrypt BLOBs in the DBMS and write them to the OS Files and invoke the respective opening program for viewing. In order to upload text files to the DBMS the user enters the File Name the Key mode (shown in the next screen) and the encryption algorithm (in the next screen). If User-specified key mode is selected the user has to enter the key in the Key Value field (in the next screen) which has to be minimum 8 characters for DES and 32 characters for AES. Then on clicking the Upload File button the file is uploaded and encrypted to a BLOB in the DB Table.

4. EXPERIMENTAL RESULTS

This section explains the different screenshot for the above said architecture. Figure-2 shows the cryptography main menu.

©2006-2015 Asian Research Publishing Network (ARPN). All rights reserved.

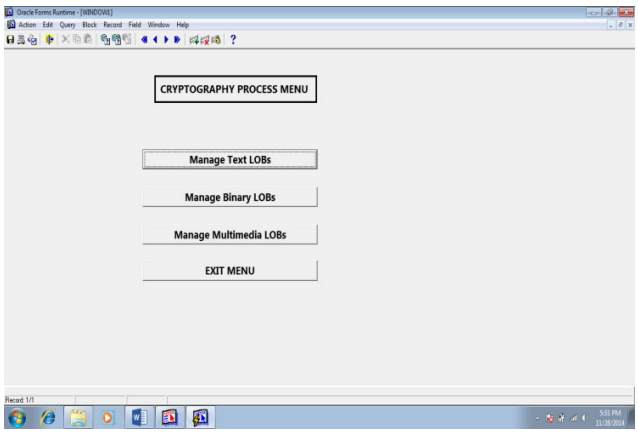

Figure-2. Cryptography process menu.

Next following figure shows the Text LOBs processing form, document LOBs processing form and multimedia LOBs processing form.

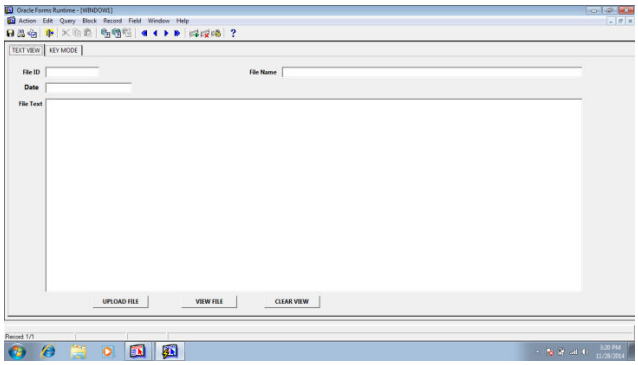

Figure-3. Text LOBs processing form.

5. CONCLUSION AND FUTURE ENHANCEMENTS

Distributed data storage schemes provide the users with convenience to outsource their files to untrusted proxy servers. Identity-based secure distributed data storage schemes are a special kind of distributed data storage schemes where users are identified by their identities and can communicate without the need of verifying the public key certificates. In this paper, we proposed two new IBSDDS schemes in standard model where, for one query, the receiver can only access one file, instead of all files. Furthermore, the access permission can be made by the owner, instead of the trusted party. Notably, our schemes are secure against the collusion attacks. The first scheme is CPA secure, while the second one is CCA secure.

REFERENCES

[1] P. Samarati and S.D.C. di Vimercati. 2010. Data Protection in Outsourcing Scenarios: Issues and Directions. Proc. ACMSymp. Information, Computer and Comm. Security (ASIACCS ’10). pp. 1-14.

[2] V. Kher and Y. Kim. 2005. Securing Distributed Storage: Challenges, Techniques, and Systems. Proc. ACM Workshop Storage Security and Survivability (StorageSS ’05). pp. 9-25.

[3] S. Jiang, X. Zhang, S. Liang, and K. Davis. 2010. Improving Networked File System Performance Using a Locality-Aware Cooperative Cache Protocol. IEEE Trans. Computers. 59(11): 1508-1519.

[4] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik. 2008. Scalable and Efficient Provable Data Possession. Proc. Int’l Conf. Security and Privacy in Comm. Networks (SecureComm ’08).

[5] H. Hacig€um€us, B.R. Iyer, C. Li, and S. Mehrotra. 2002. Executing SQL over Encrypted Data in the Database-Service-Provider Model. Proc. ACM SIGMOD Int’l Conf. Management of Data. 2002: 216-227.

[6] L. Bouganim and P. Pucheral. 2002. Chip-Secured Data Access: Confidential Data on Untrusted Servers. Proc. Int’l Conf. Very Large Data Bases (VLDB ’02). pp. 131-142.

[7] U. Maheshwari, R. Vingralek and W. Shapiro. 2000. How to Build a Trusted Database System on Untrusted Storage. Proc. Symp. Operating System Design and Implementation (OSDI ’00). pp. 135-150.

[8] G. Ateniese, K. Fu, M. Green and S. Hohenberger. 2005. Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage. Proc. Network and Distributed System Security Symp. (NDSS ’05). pp. 1-15.

[9] S.D.C. di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi and P.Samarati. 2011. Efficient and Private Access to Outsourced Data. Proc. Int’l Conf. Distributed Computing Systems (ICDCS ’11). pp. 710-719.

[10] Juels and B.S. Kalki Jr. 2007. PORs: Proofs of Retrievability for Large Files. Proc. ACM Conf. Computer and Comm. Security (CCS ’07). pp. 584-597.