ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Privacy in Offloading Data from Mobile to Cloud

Shiv Kanth B[1], Sriram N[2], Deepak S[3]

Student, Computer Science, University of Arizona, USA[1]

Programmer Analyst, Cognizant Technology Solutions, India[2, 3]

Abstract: Mobile devices namely smart phones and tablets play a significant role in life of a common man and thus they become a necessary tool in providing various services from different applications. But these devices are more limited with respect to desktop computers in terms of on-board memory, processors, and available network bandwidth. Offloading data to the high storage cloud servers invariably improves performance of mobile systems but technical challenges do remain. Moreover, the cloud based data is constantly being accessed by the mobile devices that are resource constrained for which the cost of processing must be minimised. Thus in this paper we propose a technique using attribute and re-encryption based key management with Manager as a trusted entity independent of CSP(Cloud Service Provider) in order to overcome the privacy issues.

Keywords: Security, Encryption, Decryption, Privacy, Hybrid Attribute, Data Offloading

I. Introduction

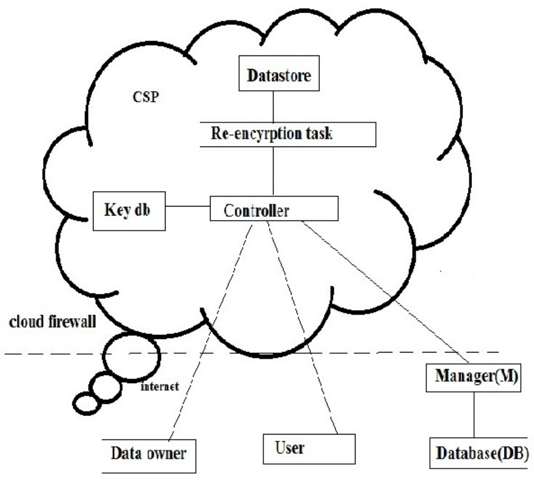
Mobile Cloud Computing binds the cloud computing into mobile environment and it significantly overcomes various obstacles related to performance in terms of battery life , bandwidth and security(reliability and privacy). Cloud based data offers numerous advantages as it enables users of mobile systems to use resources in an on-demand fashion. The rapid explosion of mobile applications and cloud computing support eases the lives of mobile users indirectly. The data that is outsourced to a cloud is appropriate for any type of applications for storage purposes and hence they are disseminated to users. Major problem that is not addressed is that the data stored in the cloud is read by the cloud service provider without the knowledge of the data owner. In addition to that, the cloud provider may not be trusted by all means and there is a considerable chance of data being intercepted by an unauthorised user despite the safety measures undertaken by the cloud provider. It is therefore necessary to use certain encryption algorithms to the data stored in the cloud in order to preserve privacy. It is also advisable that the encryption algorithms that are employed to the data of the data owner must be lightweight in terms of computation and also efficient.

A. Proposed protocol features

- This protocol is designed such that the authorized users can read the data of the data owner based on the possession of the right attributes without any arbitration from the data owner

- The responsibility of key generation is divided between data owner and a trusted authority namely Manager and thus the data owner is made free from other high computational burdens

- The security mechanism is provided through lightweight encryption algorithm such as Rijndael algorithm(AES) which is more flexible, secure and it also consumes less memory.



- Re-encryption is done in the cloud to provide additional security to the stored cipher text in the cloud and it also permits efficient revocation of users.

- This proposed protocol is demonstrated on popular mobile and cloud platforms and in addition to that scalability of this protocol is clearly addressed in terms of computational workloads.

B. Entities involved

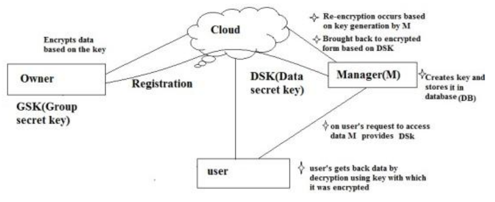
- A mobile user acting as a data owner decides what access privileges are appropriate for the data that it uploads to the cloud and retains control over a specific subset of the user population.
- A Manager is a trusted authority within the system

and is administered under the domain of the users and it is completely independent of the CSP(Cloud Service Provider).

- A cloud controller administrates access through external client interfaces. Requests, including data uploads and downloads, are made over the reliable but insecure medium of the Internet.

C. Overall Flow and Process

- New user registers.
- Manager allows and generates keys.
- User chooses the file depending on group and access permissions and encrypts file using key provided by the Manager using AES Reijndael algorithm.
- The file gets uploaded to cloud controller. Cloud controller now requests Manager for key(for re-encryption).


Cloud
Re-encryption occurs based on key generation by M
Brought back to encrypted form based on DSK
Encrypts data based on the key
Owner
Registration
DSK(Data secret key)
Manager(M)
Creates key and stores it in database (DB)
GSK(Group secret key)
on user's request to access data M provides DSk
user
user's gets back data by decryption using key with which it was encrypted

- Based on group, access and filename Manager sends a key and using the key, cloud re-encryption occurs and the re-encrypted data is stored in the cloud
- Third party user requests the Manager for the file.
- The Manager based on the group and access sends a decryption key.
- The user also requests the owner secret key from the cloud.
- Using the above two keys, the encrypted data is decrypted and plaintext is obtained.

II. LITERATURE SURVEY

A. A Study on Rijndael Algorithm for providing confidentiality to Mobile Devices

This is an era of wireless (mobile) communication and computing where mobile devices such as personal digital assis- tants, smartphones, etc., are being used in place of traditional computers. To secure wireless communication many services and protocols have been developed such as IPSec, SSL/TLS, etc. Heavy computations such as generating RSA keys (large prime numbers) are difficult to generate on mobile devices because the mobile device possesses limited resources. But it can be simplified with the help of smartcards. The paper describes how the smartcards are used to provide content encryption for mobile devices to secure the communication. The well-known Rijndael cryptographic algorithm is used in this concept to secure the communication between the mobile devices or mobile device with any conventional server.

B. Improved proxy re-encryption schemes with Applications to secure distributed storage

This paper predicts that fast and secure re-encryption will become increasingly popular as a method of managing en- crypted file systems. The usefulness of proxy re-encryption as a method to manage access controls on secure files is also demonstrated successfully. The performance measures clearly indicate that this method could be put to good use effectively in practice.

C. Trusted data sharing over untrusted cloud storage providers

The off-premises computing paradigm that comes with cloud computing has incurred great concerns on the security of data, especially the integrity and confidentiality of data, as cloud service providers may have complete control on the computing infrastructure that underpins the services. This makes it difficult to share data via cloud providers where data should be confidential to the providers and only authorized users should be allowed to access the data. This work aims to construct a system for trusted data sharing through untrusted cloud providers, to address the above mentioned issue. The constructed system can imperatively impose the access control policies of data owners, preventing the cloud storage providers from unauthorized access and making illegal authorization to access the data.

D. A Key-Policy Attribute-Based Broadcast Encryption

According to the broadcast encryption scheme with wide ap- plications in the real world without considering its security and efficiency in the model simultaneously an unbounded, Key- Policy Attribute-Based Broadcast Encryption scheme (KP- ABBE) was proposed by combining with waters dual system encryption, attribute-based encryption and broadcast encryp- tion system. Based on the standard model, the scheme can achieve constant-size public parameters, the public parameters do not impose additional limitations on the functionality of the systems (unbounded) and either a small universe size or a bound on the size of attribute sets avoid to fixed at setup.

E. Hybrid Attribute and Re-encryption based key management for scalable application in clouds.

This paper emphasizes on the fact that a third party user can obtain access to the encrypted data sent by the data owner based on the possession of certain attributes that satisfy an access structure defined in the cloud, rather than the possession of a key that must be disseminated to all interested parties in advance. The required attributes may be determined by a data owner in

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

advance. It is implemented with the help of a Manager who acts as a trusted entity independent of the Cloud Service provider and he jointly cooperates with the data owner in sending the keys.

The technique of Cipher text-Policy Attribute-Based Encryption (CP-ABE) offers numerous advantages. It allows a user to obtain access to encrypted data in the cloud based on the possession of certain attributes that satisfy an access structure defined in the cloud, rather than the possession of a key that must be disseminated to all interested parties in advance. The requisite attributes may be determined by a data owner in advance; this owner is responsible for generating the user data to be shared, encrypting it and uploading it to the cloud.

F. Designing a Hybrid Attribute-Based Encryption Scheme Supporting Dynamic Attributes

This article presents the design of a novel hybrid attribute-based encryption scheme. The scheme is attribute-based, as it allows encrypting under logical combinations of attributes, i.e. properties that users satisfy. It is hybrid, as it combines ciphertext-policy attribute-based encryption (CP-ABE) with location-based encryption (LBE) on the level of symmetric keys. It can efficiently handle dynamic attributes with continuous values, like location, even in resource-constrained settings.

III. ENCRYPTION

Cloud encryption is a service offered by cloud storage providers whereby data, or text, is transformed using encryption algorithms and is then placed on a storage cloud. Cloud encryption is the transformation of a cloud service customer's data into ciphertext. Cloud encryption is almost identical to in host encryption with one important difference the cloud customer must take time to learn about the provider's policies and procedures for encryption and encryption key management. The cloud encryption capabilities of the service provider need to match the level of sensitivity of the data being hosted.

Because encryption consumes more processor overhead, many cloud providers will only offer basic encryption on a few database fields, such as passwords and account numbers. At this point in time, having the provider encrypt a customer's entire database can become so expensive that it may make more sense to store the data in-house or encrypt the data before sending it to the cloud. To keep costs low, some cloud providers have been offering alternatives to encryption that dont require as much processing power. These techniques include redacting or obfuscating data that needs to remain confidential or the use of proprietary encryption algorithms created by the vendor.

In the past, many businesses felt comfortable allowing the cloud provider to manage encryption keys, believing that security risks could could be managed through contracts, con- trols and audits. Over time it has become apparent, however, that cloud providers cannot honor such commitments when responding to government requests for information.

IV. RIJNDAEL ALGORITHM

This algorithm is designed to be efficient both in hardware and software across a variety of platforms. Its a block cipher algorithm which works iteratively that has

- Block size: 128 bit (but also 192 or 256 bit)
- Key length: 128, 192, or 256 bit
- Number of rounds: 10, 12 o 14
- Key scheduling: 44, 52 or 60 subkeys having length 32=bits

$K_{0,0}$	$K_{0,1}$	$K_{0,2}$	$K_{0,3}$
$K_{1,0}$	$K_{1,1}$	$K_{1,2}$	$K_{1,3}$
$K_{2,0}$	$K_{2,1}$	$K_{2,2}$	$K_{2,3}$
$K_{3,0}$	$K_{3,1}$	$K_{3,2}$	$K_{3,3}$

Fig 1. Figure representing key

A. Key and Block

Represented with a matrix(array) of bytes with 4 rows and Nk columns, Nk=key length / 32. For eg, key length of 128 bits consists of Nk columns as 4. Similarly, for 4 rows and Nb columns, Nb=block length / 32 and thus for 128 bits Nb=4.Internally, the AES algorithms operations are performed on two-dimensional array of bytes called the State S. S (r,c) denotes the byte in row r and column.

In_0	In_4	In_8	In_{12}
In_1	In_5	In_9	In_{13}
In_2	In_6	In_{10}	In_{14}
In_3	In_7	In_{11}	In_{15}

Fig 2. Figure representing block

B. Subbyte Transformation

Byte substitution using a non-linear (but invertible) S-Box (independently on each byte). S-box is represented as a 16x16 array, rows and columns indexed by hexadecimal bits. 8 bytes replaced as follows :8 bytes define a hexadecimal number rc, then sr, c = binary(S-box(r,c)).

C. Shiftrows

The ShiftRows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

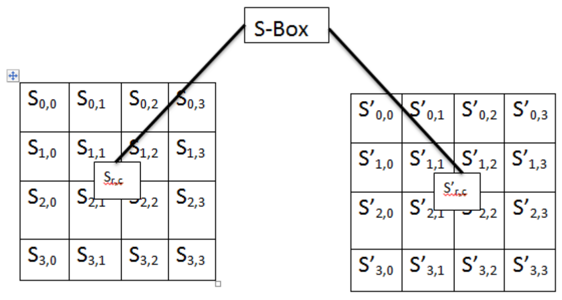


Fig 3. Figure representing Subbytes

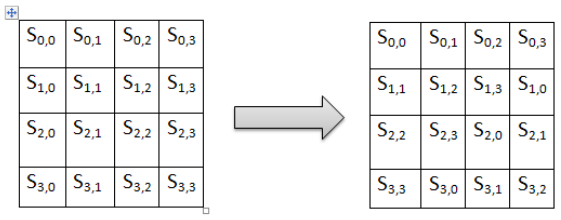


Fig 4. Figure representing Shiftrows

row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively.
For blocks of sizes 128 bits and 192 bits, the shifting pattern is the same. Row n is shifted left circular by n-1 bytes. In this way,each column of the output state of the ShiftRows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). For a 256-bit block, the first row is unchanged and the shifting for the second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks. The importance of this step is to avoid the columns being linearly independent, in which case, AES degenerates into four independent block ciphers.

D. Mixcolumns

Interpret each column as a vector of length 4.Each column of State is replaced by another column obtained by multiplying that column with a matrix in a particular field(Galois Field). Matrix multiplication is composed of multiplication and addition of the entries, and here the multiplication operation can be defined as this: multiplication by 1 means no change, multiplication by 2 means shifting to the left, and multiplication by 3 means shifting to the left and then performing XOR with the initial upshifted value.

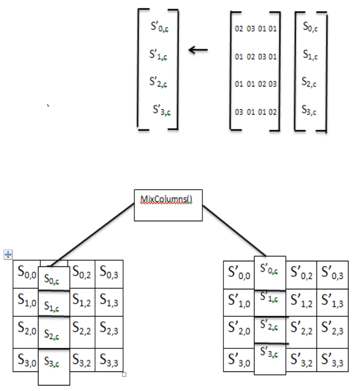


Fig 5. Figure representing Mixcolumns

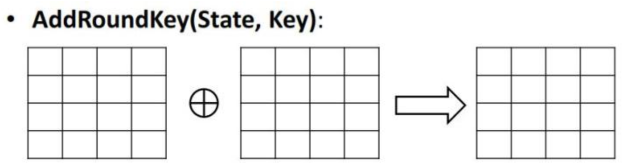


Fig 6. Figure representing AddRoundKey

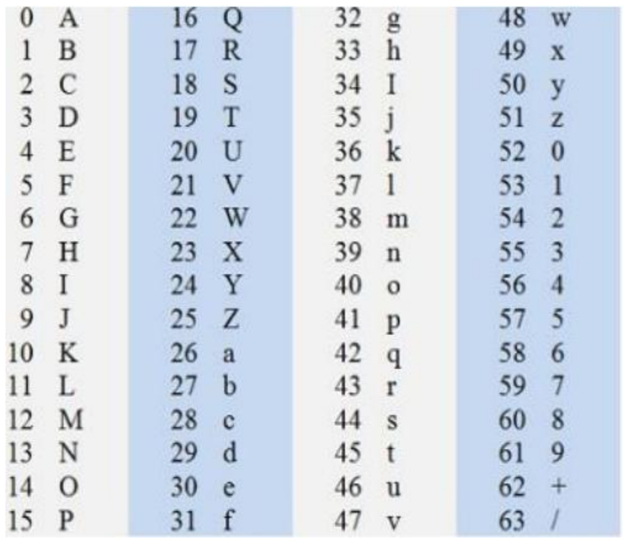
- After shifting, a conditional XOR with 0x1B should be performed if the shifted value is larger than 0xFF. (These are special cases of the usual multiplication in GF(28).)
- Addition is simply XOR.

E. AddRoundKey

In the AddRoundKey step, each byte of the state is combined with a byte of the round subkey using the XOR operation(). The subkey is combined with the state. For each round, a subkey is derived from the main key and each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.

V. BASE 64 ENCODING

Base64 is a group of similar binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radix-64 representation. The term Base64 originates from a specic MIME content transfer en- coding. Base64 encoding schemes are commonly used when there is a need to encode binary data that needs to be stored and transferred over media that is designed to deal with textual data. This is to ensure that the data remains intact without modication during transport. Base64 is commonly used in a number of applications including email via MIME, and storing complex data in XML.



0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	I	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Fig 7. Figure representing character mapping

Base 64 encodes the input data three bytes at a time. Each block of three input bytes is encoded to create a block of four printable characters. The three bytes are ordered into a 24 bit value, starting from the most signicant bit of Byte 0, and ending with the least signicant bit of Byte 2. The bits are then arranged as a set of four 6 bit numbers, N0 to N3. The rst 6 bits form N0, the next 6 form N1 etc. Each 6 bit number has a range 0 to 63.

The second stage is to convert numbers N0N3 into ASCII characters, C0C3. This is done according to Fig. 7

In addition, the MIME specification states that encoded data should have a CRLF character pair inserted after every 76 characters (or less) of encoded data. The original data can be any length, not necessarily a multiple of 3. This means that the last block of binary data could be 1, 2 or 3 bytes long. To code it, we add zeros to the nal block to make it a multiple of 3, and convert it to 4 characters as usual. However, we indicate the length of the block in the following way: If the nal block has a length of 1 byte, the encoded characters consist of C0, C1, followed by 2 = characters (C2 and C3 contain no useful information anyway). If the nal block has a length of 2 bytes, the encoded characters consist of C0, C1, C2, followed by a single = character. If the nal block has a length of 3 bytes, the encoded characters consist of C0, C1, C2, C3 in then normal way.

The encoded value of Man is TWFu. Encoded in ASCII, the characters M, a, and n are stored as the bytes 77, 97, and 110, which are the 8-bit binary values 01001101,01100001 and 01101110. These three values are joined together into a 24-bit string, producing 010011010110000101101110. Groups of 6 bits (6 bits have a maximum of 26=64 different binary values) are converted into individual numbers from left to right (in this case, there are four numbers in a 24-bit string), which are then converted into their corresponding Base64 character values.

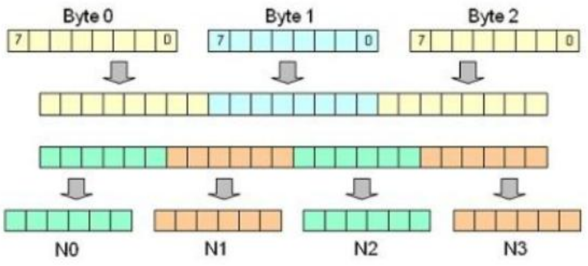
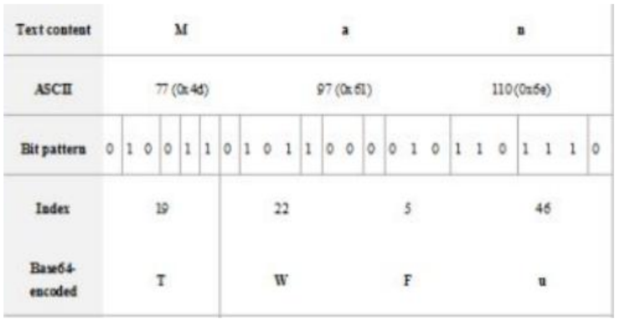


Fig 8. Figure representing Byte Process



Text content	M	a	n	
ASCII	77 (0x4d)	97 (0x61)	110 (0x6e)	
Bit pattern	0 1 0 0 1 1 0 1	0 1 1 0 0 0 0 1	0 1 1 0 1 1 1 0	
Index	19	22	5	46
Base64-encoded	T	W	F	u

Fig 9. Figure representing an example of Base64.

VI. IMPLEMENTATION

The proposed protocol was implemented and proled to gauge its performance. It was realized on popular existing commercial platforms, including the Google Android mobile and the open source eyeOS cloud platform.

A. *Software requirements*

- Windows Operating System 2000 and Above
- JDK 1.6
- Tomcat Server 6.0
- My SQL 5.5
- Android mobile
- XAMPP

B. *Hardware requirements*

- Hard Disk: 10GB and Above
- RAM: 512MB and Above
- Processor: Pentium III and Above

C. *Cloud Platform*

eyeOS is a web desktop following the cloud computing concept that seeks to enable collaboration and communication among users. It is mainly written in PHP, XML, and JavaScript. It is a cloud application platform with a web-based desktop interface. Commonly called a cloud desktop because of its unique user interface, eyeOS delivers a whole desktop from the cloud with le management, personal management information tools, collaborative tools and with the integration of the clients applications.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

VII. CONCLUSION AND FUTURE WORK

- A key management system has been proposed for secure data outsourcing applications, whereby attribute based encryption effectively permits authorized users to access secure content in the cloud based on the satisfaction of an attribute based policy.
- The data owner and a trusted authority co-operate in key generation and encryption processes such that intensive cryptographic operations are minimised for the data owner.
- The above protocol could be expanded for applications with greater data values such as media files, streaming, money based transactions, client-server communication.

ACKNOWLEDGEMENT

We would like to thank and extend our deep sense of gratitude to our **Professor Mr. N. Sujaudeen**, Department of Computer Science and Engineering and our institution, SSN College of Engineering, Chennai for their constant support that helped us to shape and refine our work throughout this project.

REFERENCES

[1] 1. Fahad Bin Nafey and OBV Ramanaiah. (2008) A Study on Rijndael Algorithm for providing Confidentiality to Mobile Devices,in Region 10 IEEE Conference in TENCON

[2] 2. G.Ateniese, K. Fu, M. Green, and S. Hohenberger. (2006) Improved proxyre-encryption schemes with applications to secure distributed stor- age, ACM Transactions of Information and System Security, vol. 9, pp.130

[3] 3. G.Zhao, C. Rong, J. Li, F. Zhang, and Y. Tang. (2010) Trusted data sharing over untrusted cloud storage providers, in Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science, ser. CLOUDCOM 10. Washington, DC, USA: IEEE Computer Society.pp. 97103

[4] 4. P. K. Tysowski and M. A. Hasan. (2011) Towards Secure Commu- nication for highly Scalable Mobile Applications in Cloud Computing Systems, Centre for Applied Cryptographic Research (CACR), University of Waterloo, Tech. Rep. 33.

[5] 5. Foster, C., Uchitel, C., Magee, J. and Kramer J. (2003) Model-based verication of web service compositions, In proc. of ASE03, pp. 152-163.

[6] 6. Q. Liu, G. Wang, and J. Wu. (2012) Clock-based proxy re-encryption scheme in unreliable clouds, in Parallel Processing Workshops (ICPPW) 41st International Conference on, sept. 2012, pp. 304 305 .

[7] 7. S. Jahid, P. Mittal, and N. Borisov. (2011) EASiER: encryption- based access control in social networks with efficient revocation, in Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, ser. ASIACCS 11 New York, NY, USA: ACM, pp. 411415