An Efficient and Secure Protocol for Ensuring Data Storage Security over Cloud

Dheeraj Gupta[1] Ajay M Jethwa[2] Ravikumar Kamma[3] Shreyank Joshi[4] Prof. Deepali Maste[5]
[1,2,3,4]Student [5]Assistant Professor
[1,2,3,4,5]Department of Computer Engineering
[1,2,3,4,5]Atharva College of Engineering, Mumbai University,Mumbai, India

Abstract— The previously established protocols often fail as a tool for ensuring strigent security of users. In this paper, we propose an efficient and secure protocol to address these issues. Our design is based on Elliptic Curve Cryptography. Our method facilitates a user to periodically verify the data integrity of the ongoing uplink or downlink connectivity prior to revelation of the original data. Once, authenticated the user proceeds with accessing the data.

Key words: Efficient and Secure Protocol, Security over Cloud

I. INTRODUCTION

With the invention of Cloud Storage, the days of keeping all your documents, photos, music files etc. on your computer are long gone. Today, the cloud storage is fulfilling the need for extra storage space via means of a virtual storage to hold all of your digital data. Cloud Storage space providers operate large data centers, and provide people with additional space by hosting the required data. They can either buy or lease storage capacity from these data centers suiting their needs. The data center operators, in the background, make the resources virtual according to the requirements of the customer and expose them as storage pools of data, accessible only to the customers themselves in order to store additional files or access data objects. Physically, these resources can stretch across multiple different servers.

II. LITERATURE SURVEY

Cloud storage is a model of networked enterprise storage where data is stored not only in the user's computer, but in virtualized pools of storage which are generally hosted by third parties.

1) Ravi Shankar Dhakar in the "Modified RSA Encryption Algorithm (MREA)" talks about factorization in RSA cryptosystem, and their implementation compares the existing system and their system with key sizes up to 1024 bit. The authors claim their system to be better than existing system for the brute-force attack.
2) Suli Wang talks about the "File encryption and decryption system based on RSA algorithm" where they used RSA for encryption and decryption of files with smaller sizes.
3) Maryam Savari in "Comparison of ECC and RSA Algorithm in Multipurpose Smart Card Application" compare the security of RSA 1024-bit key versus ECC 160- bit key sizes[2].
4) P.R. Vijayalakshmi in "Performance Analysis of RSA and ECC in Identity-Based Authenticated New Multiparty Key Agreement Protocol" compare ECC algorithm with 128 bits with that of RSA algorithm with 1024 bits key size.
5) Kamlesh Gupta in "ECC over RSA for Asymmetric Encryption: A Review" demonstrated the use ECC for portable devices and applications [7].
6) Arjun Kumar proposes a method that allows user to store and access the data securely from the cloud storage in "Secure Storage and Access of Data in Cloud Computing".
7) This work compares the security of ECC in the key range of 160 - 512 bits and RSA key sizes ranging from 512 - 3072 bits. The simulation experiments compare the ECC and RSA at different levels of key sizes and block sizes.

III. EXISTING SYSTEM

RSA is one of the first successfully implemented public-key cryptosystems and is widely used for secured transmission over a vulnerable communication channel. In such a cryptosystem, the encryption key is public while the decryption key is kept secret. In RSA, the asymmetry is based on the mathematical difficulty of factoring the product of two large prime numbers, often termed as the factoring problem. RSA was initially proposed and implemented by Ron Rivest, Adi Shamir, and Leonard Adleman, in 1977.[11] Hence, the name is derived from the initials of each author. RSA algorithm is the most commonly used asymmetric/public key cryptography algorithm for encryption and decryption by various cloud service vendors today. Being a first generation algorithm that was used for providing data security, it can be used to encrypt a message without having to exchange a secret key separately. The RSA algorithm can be used for both encryption and authentication by method of digital signatures. The security of RSA is based on the difficulty of factoring large prime integers. Party A sends an encrypted message to party B without any prior exchange of key and B decrypts it using its own private key, which only he knows. As mentioned RSA can also be used to sign a message, so A can sign a message using his private key and B can verify it using A's public key which is publicly available. Encryption of a message, say m, involves exponentiation, c = m^e mod n, which requires a lot of

mathematical computations. In RSA cryptosystem, user A picks up two large primes p and q, computes their product, n = p*q. Now A's public key is a pair of integers {n, e} and the private key is d.[7]

IV. PROPOSED SYSTEM

Elliptical curve cryptography (ECC) is a public key encryption technique based on elliptic curve theory that can be used to create faster, smaller, and more efficient cryptographic keys. ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers. Because ECC helps to establish equivalent security with lower computing power and battery resource usage, it is becoming widely used for mobile applications. ECC was developed by Certicom, a mobile e-business security provider, and was recently licensed by Hifn, a manufacturer of integrated circuitry (IC) and network security products.

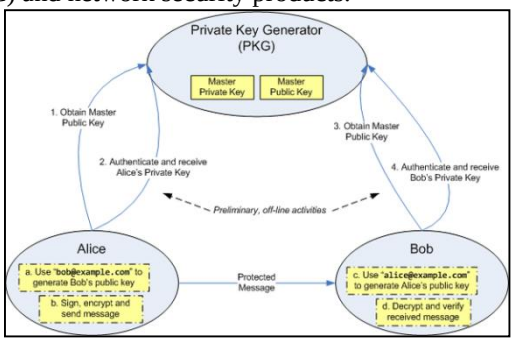

Fig. 1: Overview of the Approach

The properties and functions of elliptic curves have been studied in mathematics for 150 years. Their use within cryptography was first proposed in 1985, (separately) by Neal Koblitz from the University of Washington, and Victor Miller at IBM. An elliptic curve is not an ellipse (oval shape), but is represented as a looping line intersecting two axes (lines on a graph used to indicate the position of a point). ECC is based on properties of a particular type of equation created from the mathematical group (a set of values for which operations can be performed on any two members of the group to produce a third member) derived from points where the line intersects the axes. Multiplying a point on the curve by a number will produce another point on the curve, but it is very difficult to find what number was used, even if you know the original point and the result. Equations based on elliptic curves have a characteristic that is very valuable for cryptography purposes: they are relatively easy to perform, and extremely difficult to reverse.

V. GENERAL TERMS

A. ELLIPTICAL CURVES

An elliptic curve is a nonsingular projective algebraic curve over some field k with genus 1 and a specified point O (this will be the "point at infinity"). So long as k does not have characteristic 2 or 3, this will be a smooth plane cubic curve with the point at infinity, and we can describe the curve as points satisfying the equation.

$y2 = x3 + ax + b$

where, a and b such that the discriminant,

$\Delta = -16\ (4a3 + 27b2)$,

is nonzero (which will give the desired non singularity).

The operation exploited for key selection in elliptic curve cryptography comes from considering the elliptic curve as an abelian group with points as elements. The group law is point addition; to add two points P and Q, we will draw the line PQ through them (or use the tangent line at P to add it to itself), find the third point of intersection −R of that line, and reflect it over the axis of symmetry of the curve. The resulting point, R, will be the sum of P and Q. For the purposes of this addition, note that the point at infinity O lies on any line through a point and it's opposite. The formal properties of the addition law are described below.

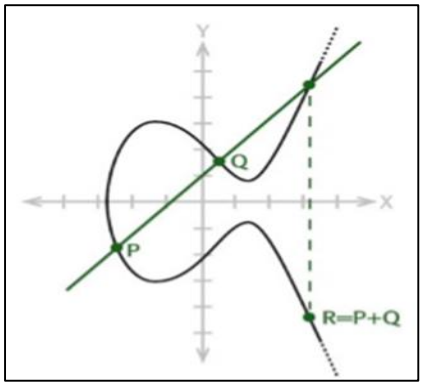

Fig. 2: Elliptical Curve Concept of Key Selection

B. KEY GENERATION

Step 1:
//For user A
PUB = G*P
UA = (PUA, PA) // User A key pair
Step 2:
// For User B
PUB = BP*PB
UB = (PUB, PB) //User B key pair
// BP is the Base Point
Step 3: //Send the Public key of UB to UA
Send (PUB, UB);
Step 4: //Send the Public key of UA to UB
Send (PUA, UA);

C. ENCRYPTION

Step 1: Calculate APL = p*AP; //p = Ascii value of text
//AP: random point on EC
Step 2: // Calculate kBP
kBP = k*BP
//BP is the Base Point
Step 3:
// Send Cipher test to receiver, i.e. User B
Cipher Text, CM = {kBP, APL+ k * PUB}

D. DECRYPTION

Let kBP be the first point
APL+ kPUB be second point
Step 1: Calculate PBkBP = PB * first_point
//this yields us an equivalent point to kPUB
Step 2: Calculate APL = (APL+ k*PUB) - PBkBP
Now using discrete logarithm concept
Step 3: Evaluate value of sent text from APL
APL = rAP
//r is the value to be calculated using the discrete logarithmic concept. r = p, i.e. the original ASCII value.

VI. DIFFI-HELLMAN KEY EXCHANGE

Diffie–Hellman key exchange (D–H) is a specific method of securely exchanging cryptographic keys over a public channel and was one of the first public-key protocols as originally conceptualized by Ralph Merkle. D–H is one of the earliest practical examples of public key exchange implemented within the field of cryptography. Traditionally, secure encrypted communication between two parties required that they first exchange keys by some secure physical channel, such as paper key lists transported by a trusted courier. The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

VII. IMPLEMENTATION

This section describes the overall methodology adopted in the project. The goal of our paper is to use ECC algorithm in order to provide better security for data exchange over a cloud server. The main feature that our project dedicatedly serves is encryption and decryption of the files while uploading and downloading to and from the Cloud Storage. Once a user is registered the framework verifies the user every time when an attempt to upload or download a file is made. This verification is done on an internal level sustaining the transparency of authentication meanwhile hiding it from the user. The authentication rights to verify a user are only accessible by the system administrator. Post verification the user can proceed to either download or upload a file depending upon the requirement. The real work begins when the file transfer to and fro from the cloud in initiated. The algorithm encrypts and decrypts the file while uploading and downloading respectively using ECC and generating key for both the sender and receiver on a asymmetric basis. Hence, the entire methodology works in a rather chronological order of user authentication, key generation, encryption and decryption of the file over the cloud while upload and download.

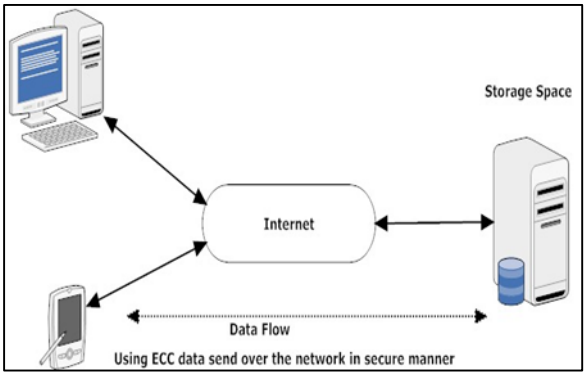

Fig. 3: System Architecture

VIII. CONCLUSION

Elliptic Curve Cryptography provides better secured and greater security and is more efficient in terms of performance than the first generation algorithm techniques like RSA currently in use. With emergence of ECC vendors should seriously consider upgrading their systems. ECC provides outstanding computational and bandwidth advantages at low cost. Although ECC's security has not been completely evaluated, expectations regarding its widespread usage in various fields in the future have exponentially grown. After comparing the RSA and ECC ciphers, the ECC has proved to involve much less overheads compared to RSA. The ECC has many advantages due to its ability to provide the same level of security as RSA yet using shorter keys.

REFERENCES

[1] JavaTM Cryptography Extension (JCE), Reference Guide. http://docs.oracle.com/javase/1.5.0/docs/guide/security/jce/JCERefGuide.html
[2] Berta, I.Z., and Z. A. Mann. "Implementing Elliptic Curve Cryptography on PC and Smart Card", Periodica Polytechnica Ser. El. Eng. Vol 46. NO 1-2, PP 47. 2002.
[3] Brown, M., D. L. Hankerson, J. Lopez and A. Menezes, "Software implementation of the NIST Elliptic curves over prime fields". In Progress in Cryptology - CT-RSA, D. Naccache, Ed, vol. 2020 of Lecture Notes in Computer Science, pp. 250-265. 2001.
[4] Neal Koblitz, Alfred J. Menezes, "A Survey of Public-Key Cryptosystems". Advanced Computing &Communication Technologies (ACCT), Second International Conference, 2012.
[5] Certicom Corp. "An elliptic curve cryptography (ECC) primer". White paper, Certicom. 2004.
[6] Rabah, K. "Implementation of Elliptic curve Diffie-Hellman and EC Encryption schemes" in Information technology journal, 01/2005.
[7] Rabah, K. "Implementing Secure RSA Cryptosystem Using Your Own Cryptographic JCE Provider". Journal of Applied Science, vol. 6, Issue 3, p.482-510. 2006.
[8] Robshaw, M. J. B. and Y. L. Yin. "Elliptic Curve Cryptosystems". 1997 http://www.rsasecurity.com/rsalabs/ecc/ellipticcurve.html
[9] Stallings, W. "Cryptography and Network Security: Principles and Practice, 3rd edition", Prentice Hall, New Jersey, 2003.
[10] Trappe. W and L. C. Washington "Introduction to Cryptography with Coding Theory", Prentice Hall, New Jersey, 2002.
[11] Weil, N. (). "U.S. govt.'s encryption standard cracked in record time". Network World. 1998, http://www.networkworld.com/news0720des.html
[12] Amara, M.; Lab. LAGA, Univ. Paris-8, St. Denis, France; Siad, A. "Elliptic Curve Cryptography and its applications".