Security in Cloud using Cryptography – A Pivotal Approach

Presented by

Gada, Sharanya – Team Lead

Alapati, Ravindranath Chowdary

Dasari, Venkata Reddy

Palem, Kishore

Perumalla, Venkata Krishna Meher

Shaik, Khadar

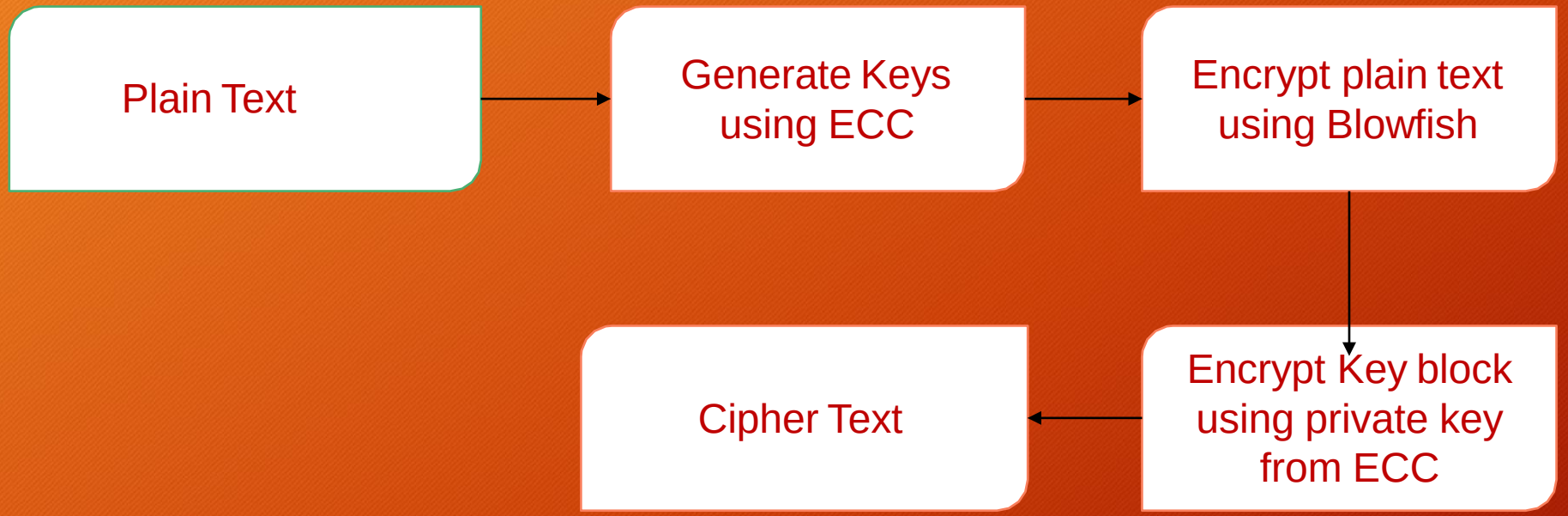
Agenda

- Cloud Computing
- Cryptography
- Elliptical Curve Cryptography
- Blowfish
- Outcomes
- Conclusion

Cloud Computing

- Cloud computing is an Internet-primarily based computing version that affords numerous sources via Cloud Service Providers (CSP) to Cloud Users (CU) on call for foundation without shopping for the underlying infrastructure and follows a pay-per-use foundation.
- It helps with virtualization of physical resources if you want to improve performance and accomplish more than one duty at the same time.
- Cloud Computing Environments (CCEs) provide a variety of deployments to symbolize multiple clouds owned by an organization or institution.

Project Walkthru



Elliptic Curve Cryptography

- Elliptical Curve Cryptography is considered as one of the modern public-key cryptosystems based on algebraic structures of elliptical curves over a finite field.
- It follows the asymmetric cryptosystems such as encryption, signatures, and key exchange.

Keys in ECC

- Private keys in the ECC are in the range of the Elliptical Curve size which are 256-bit integers.

- Private key: 0x51897b64e85c3f714bba707e867914295a1377a7463a9dae8ea6a8b914246319

- Public keys in ECC are EC coordinate points(x,y) laying on the curve.
- They are compressed to just one coordinate (odd or even). Overall public key is 257-bit integer, and its corresponding private key is 256-bit.

Generator in ECC

- The ECC cryptosystems define a special pre-defined (constant) EC point called generator point G (base point), which can generate any other point in its subgroup over the elliptic curve by multiplying G by some integer in the range [0...**r**].
- The number r is called "order" of the cyclic subgroup (the total number of all points in the subgroup).

- Finally, in the ECC cryptography the EC points, together with the generator point **G** form cyclic groups (or cyclic subgroups), which means that a number **r** exists (**r** > 1), such that **r** * **G** = 0 * **G** = ***infinity*** and all points in the subgroup can be obtained by multiplying **G** by integer in the range [1...**r**]. The number r is called order of the group (or subgroup).

ECC Walkthru

- In ECC, you can get EC point P (corresponding public key) by multiplying fixed EC point G (generator point) by a specific integer k (k can be considered a private key).

- Consequently, in ECC we have:

  Elliptic curve (EC) over finite field $\mathbb{F}_p$
  G == generator point (fixed constant, a base point on the EC)
  k == private key (integer)
  P == public key (point)

Real time ECC Curves

Elliptic curves in the elliptic curve cryptography (ECC) may be presented in several **forms**

Weierstrass form of elliptic curve

- y2 = x3 + **a**_x + ***b***
- Example Weierstrass curve used in ECC is secp256k1, which has the form y2 = x3 + ***7***

- **Montgomery form** of elliptic curve:
  - **_B**_y2 = x3 + **_A**_x2 + x
  - Example Montgomery curve used in ECC is Curve25519, which has the form y2 = x3 + **_486662**_x2 + x

- **Edwards form** of elliptic curve:
  - x2 + y2 = 1 + **_d**_x2y2
  - Example Edwards curve used in ECC is Curve448, which has the form x2 + y2 = 1 - **_39081**_x2y2

Example

- Private Key:

  b'8175f7cd524a59b6efbd447985ce5d97c546b319521ff236203970e50052c6412

- Public Key:

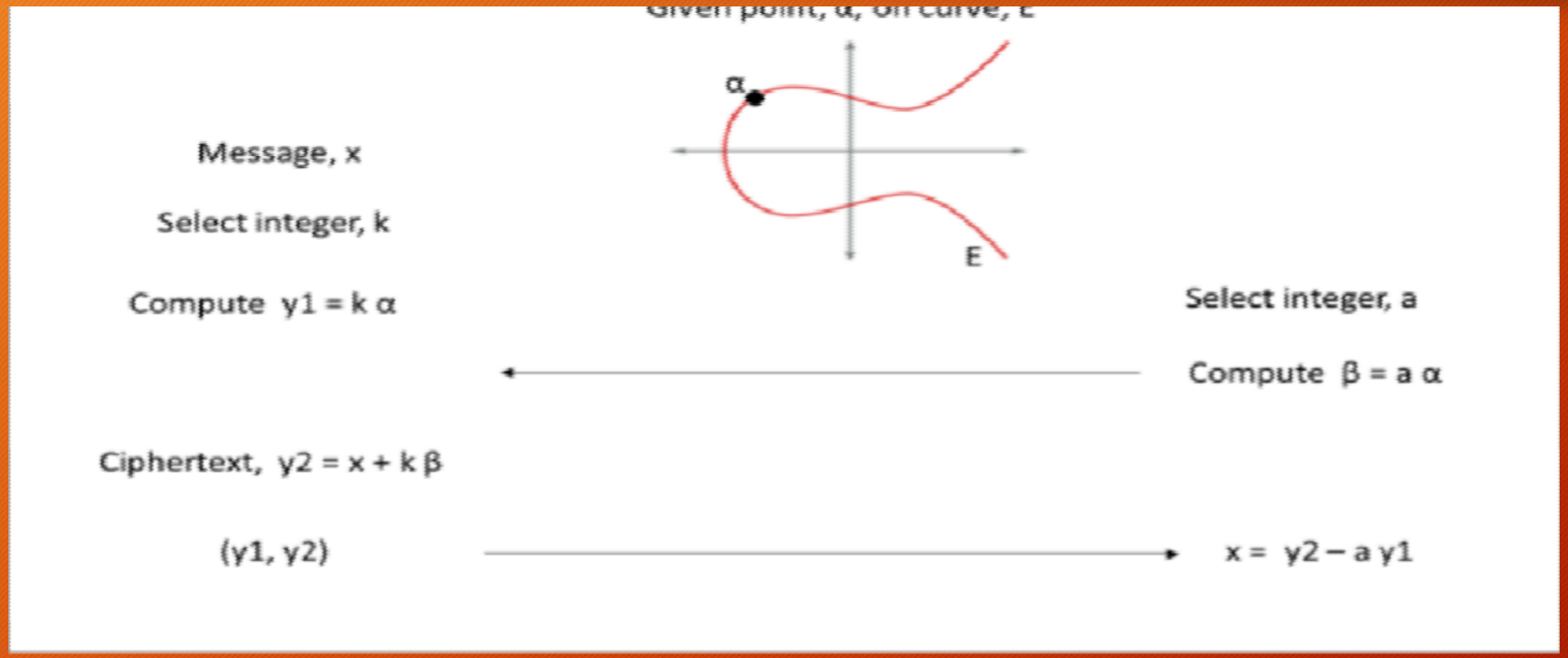
  b'cf97a96568fee4ddb232f617fd5b9df2d2e5b90e68ba7f6d5129ea92d7d8f95e

Advantages

- It is very fast to calculate P = k * G, using the well-known ECC Multiplication algorithm in time log_2(kw_), e.g. the "double and add algorithm". For 256-bit curves, it will take just a few hundred simple EC operations.
- Calculating k = P / G is very slow (it is considered infeasible if k is large).

- Smaller keys and signatures than RSA
- Security
- Fast key generation,
- Fast key agreement
- Fast signatures.

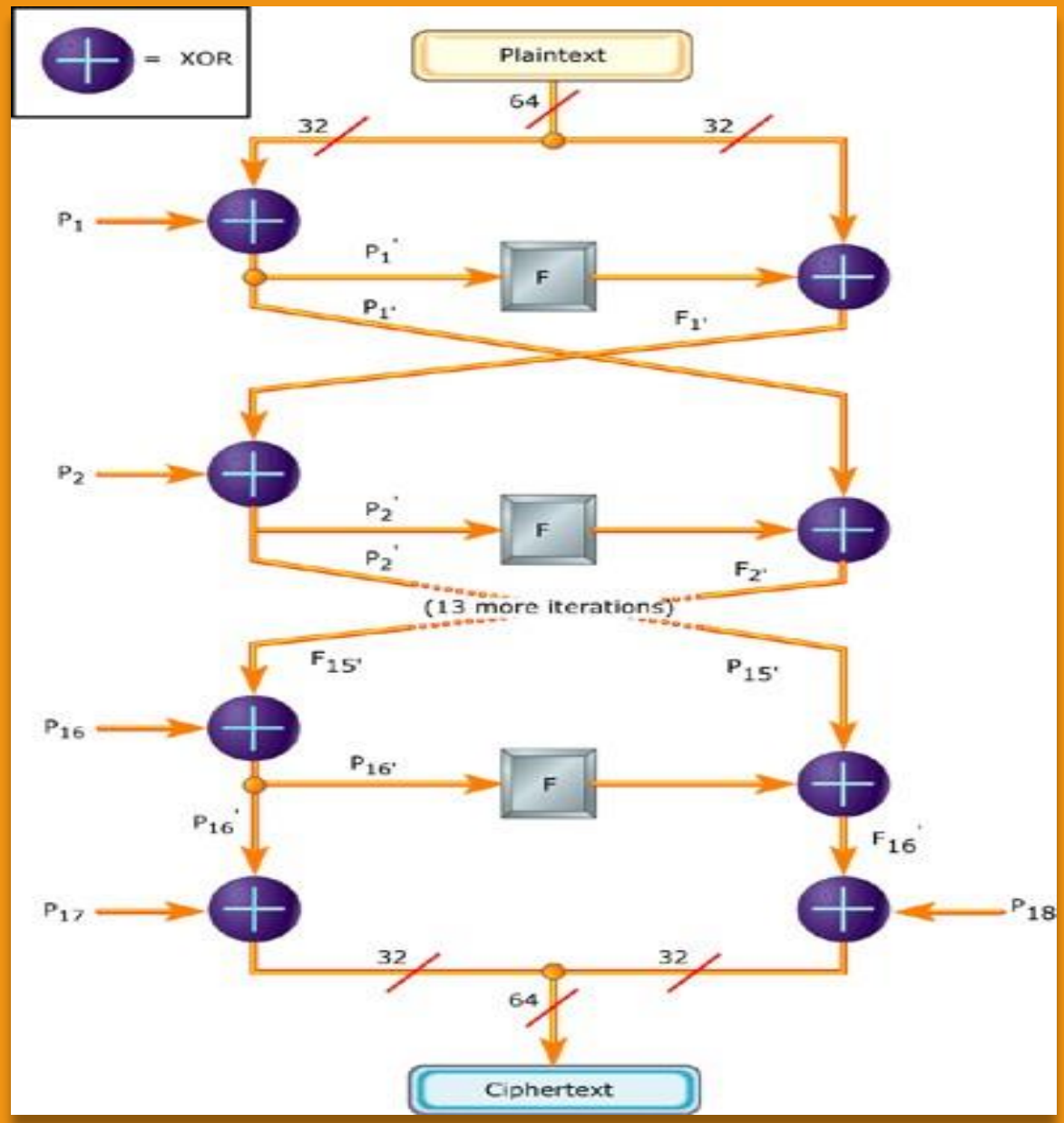
ECC Pictorial Reprsentation



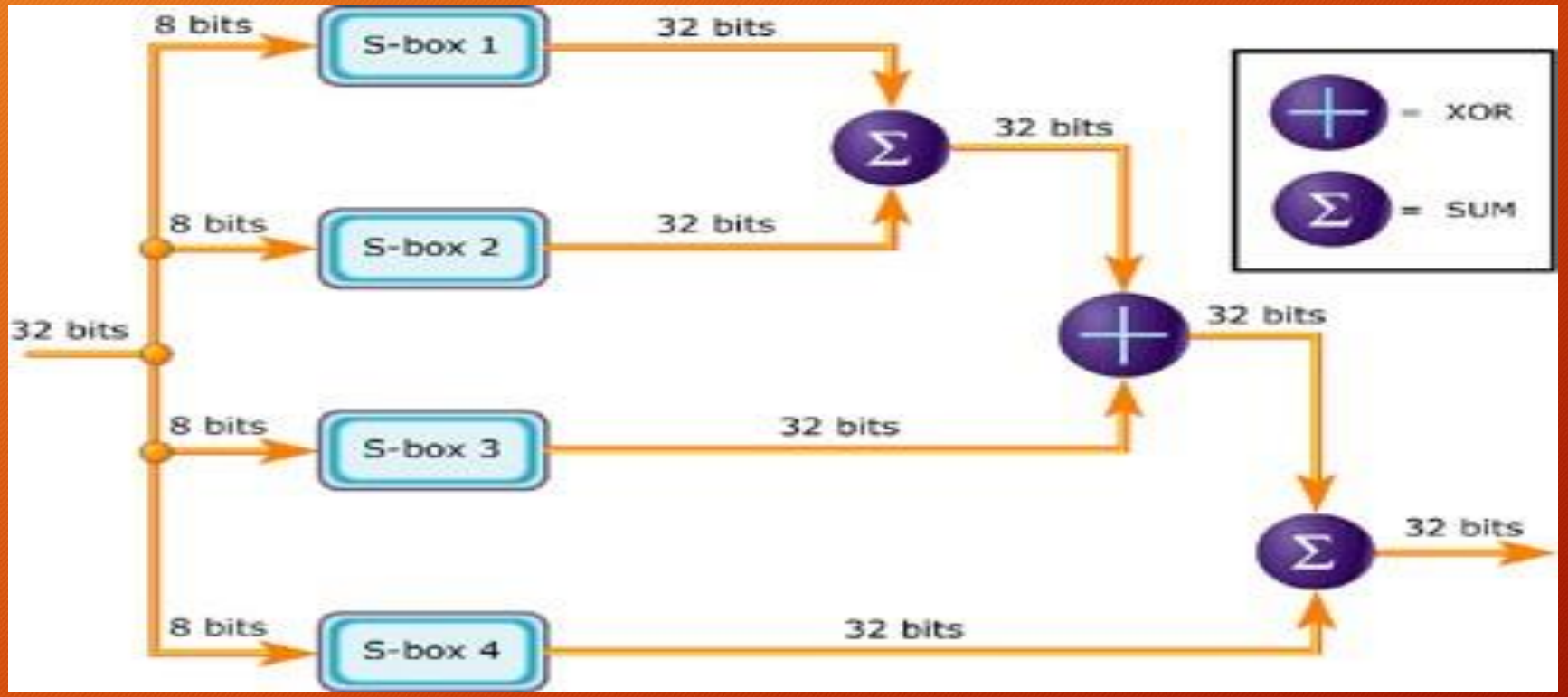
Blowfish

- Blowfish was designed in 1993 by Bruce Schneier as a fast, free alternative to existing encryption algorithms.
- Blowfish is a symmetric encryption algorithm, meaning that it uses the same secret key to both encrypt and decrypt messages. Blowfish is also a block cipher, meaning that it divides a message up into fixed length blocks during encryption and decryption. The block length for Blowfish is 64 bits; messages that aren't a multiple of eight bytes in size must be padded.

- Blowfish is a symmetric encryption algorithm.
- The block length for Blowfish is 64 bits.
- It takes a variable-length key, from 32 bits to 448 bits



Detailed View



Requirements:

- Blowfish requires about 5KB of memory. A careful implementation on a 32-bit processor can encrypt or decrypt a 64-bit message in approximately 12 clock cycles.
- Longer messages increase computation time in a linear fashion; for example, a 128-bit message takes about (2 x 12) clocks. Blowfish works with keys up to 448 bits in length.

Products that use Blowfish

- **Password Management**:
  - Access Manager
  - Java PasswordSafe
  - Web Confidential
- **File/Disk Encryption**:
  - GnuPG
  - Bcrypt
  - CryptoForge

- **Backup Tools**:
  - Symantec NetBackup
  - Backup for Workgroups
- **Email Encryption**:
  - A-Lock
  - SecuMail
- **Operating System Examples**:
  - Linux
  - OpenBSD
- **Secure Shell (SSH)**:
  - OpenSSH
  - PuTTY

Blowfish Vs AES

- Blowfish and AES are both symmetric encryption algorithms meaning both encryption and decryption keys are the same. This also means that the same key is shared to enable secure communication.
- This type of encryption is typically used for bulk data encryption. It also can be easily implemented by hardware. The main issue with symmetric encryption is that a person with the decryption key can decrypt all of the data.
- Blowfish works fast due to its bulk encryption and decryption. Blowfish uses a block size of 64 bits. It is even faster than AES implemented in software, but still, it is not as effective as AES.

Conclusion

- In this project we use ECC to generate public and private keys.
- Encrypt the plain text using Blowfish and key block with Ecc public key of the receiver.
- Compare the performance of Blowfish with AES.

References

- B, S. (2011). Cloud Computing Bible. 1st ed. Wiley.
- Bokefode , J., Ubale , S., Pingale , S., Karane , K., & Apate , S. (2015, May). Developing Secure Cloud Storage System by Applying AES and RSA Cryptography Algorithms with Role bases Access Control Model. International Journal of Computer Applications, 118.
- Kumar, A., Hoonjae, L., Byung, G., & Anu. (2012). SECURE STORAGE AND ACCESS OF DATA IN CLOUD COMPUTING. IEEE, 336339.
- M.Vijayapriya. (n.d.). security algorithm in cloud computing: overview. International Journal of Computer Science & Engineering Technology (IJCSET), 4.
- Ren, Kui, Cong , W., & Qian , W. (2012). SECURITY CHALLENGES FOR THE PUBLIC CLOUD. Internet Computing, IEEE 16.1 , 69-73.
- Sanjoli, S., & Jasmeet , S. (n.d.). Cloud computing security using encryption technique. IJARCET, 2(7).