SEMESTER -1
M.SC. COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF DELHI

CPKI
DESIGN AND IMPLEMENTATION

INFORMATION SECURITY
Under guidance of: **DR. OM PAL**

AMBEDKAR (11)
ARYAN (15)
HARSH (20)
NACHIKET (35)
PRADEEP (39)
ROHAN (47)

META-PKI

This paper proposes an algorithm for verification of public key certificates in a decentralized public key infrastructure – Meta-PKI.

REFERENCE

Simplifying Dynamic Public Key Certificate Graph for Certification Path Building in Distributed Public Key Infrastructure
Shohei Kakei, Yoshiaki Shiraishi, Shoichi Saito

MOTIVATION

A mesh PKI can eliminate single point of failure because the responsibility of a single CA is distributed across multiple CAs.

However, this is likely to complicate certification path building to determine the order in which certificates are to be verified.

The paper addresses this issue and gives an algorithm to simplify certification path building.

PROPOSED ALGORITHM

The proposal consists of 2 main parts:

1. Graph partitioning

2. Certification path building

Concept: the building certification paths problem can be solved as the shortest path problem between two public key certificates.

GRAPH PARTITIONING

a. Hash value of a given public key certificate is computed as a fingerprint.
The vertex of the fingerprint is added to the set of available vertices in
the graph.

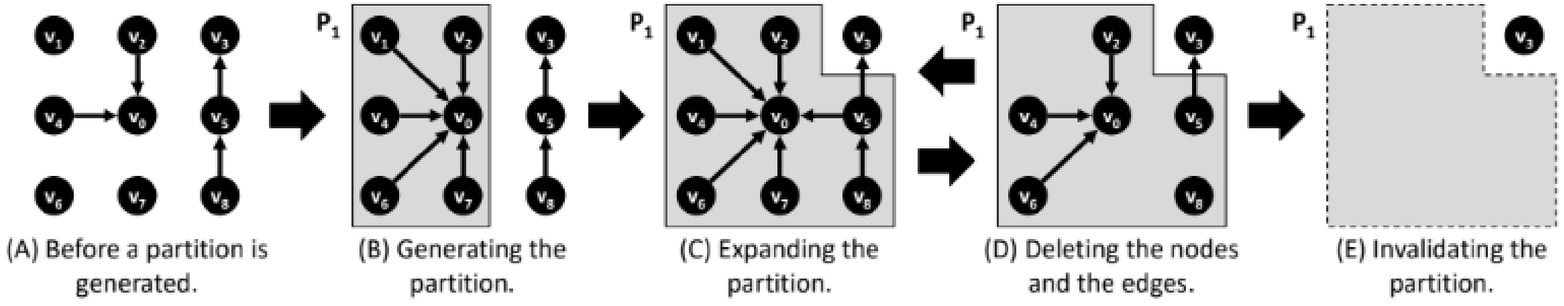


Fig. 6. Lifecycle of a partition.

GRAPH PARTITIONING

a. Hash value of a given public key certificate is computed as a fingerprint.
The vertex of the fingerprint is added to the set of available vertices in
the graph.

b. **Cross-certifying step**: an edge is connected from a source vertex to a target vertex with some weight.
5 possible cases are handled depending upon the partitions in which these vertices are present.

The shortest distance between the source vertex and the target vertex is calculated.

CROSS CERTIFYING STEP

Case 1: **Both vertices belong to the same partition.** The algorithm computes the shortest paths from each ingress to each egress vertices using some single source shortest path algorithm.

Case 2: **Both vertices belong to different partitions.** The algorithm updates the ingress and egress vertices of the partitions and computes the shortest paths.

Case 3: **Only the target vertex belongs to a partition.** The algorithm recursively follows edges in the opposite direction of source vertex andjoins the found vertices into the partition of the target vertex (the partition in which the target vertex belongs).

CROSS CERTIFYING STEP

Case 4: **Only the source vertex belongs to a partition.** The algorithm
generates a new partition around the target vertex and joins the target
vertex to the new partition. The shortest paths of source vertex's partition are updated if the source vertex becomes an egress vertex.

Case 5: **Neither vertex belongs to any partition.** The algorithm performs both case 4 and case 3 processes.

CERTIFICATION PATH BUILDING:

This algorithm takes two fingerprints of a source vertex and a target vertex and computes top-k shortest paths (KSP) using the precomputed shortest paths within each partition.

Since there can be multiple certification paths, the proposed method gives the choice of which path to use by presenting the top-k shortest certification paths.

META-PKI : ADVANTAGES

In the case of building certification paths with the naive method (Yen's algorithm), the processing time grew exponentially, while the increase of the processing time is kept constant in the proposed method.

The proposed algorithm can compute the entire certification paths more efficiently that the naive method.

META-PKI : DISADVANTAGES

1. In the proposed algorithm, the authors suggest a threshold for graph partitioning. However, no further discussion is made on the optimal choice of the said threshold.

META-PKI : DISADVANTAGES

2. Due to constraints and policies for cross-certification, some public key certificates may be considered invalid even if these certificates are in certification paths.

To make certification path building flexible, it is conceivable that all certification paths are presented to a certificate verifier, and the verifier chooses whether or not to accept the certification paths by the constraints and the policies.

However, the processing load on PKI clients will be high, and the management of the constraints and the policies of the PKI clients can be complicated.

PKI4IOT

This paper proposes an automated certificate enrolment protocol for use with IoT devices – PKI4IoT – public key infrastructure for internet of things.

REFERENCE

PKI4IoT: Towards public key infrastructure for the Internet of Things
Joel Höglunda, Samuel Lindemer , Martin Furuhedb, Shahid Razaa

MOTIVATION

1. **Security risk**: The IoT presents great security risks owing to the scale in which it is implemented. Lack of authentication mechanisms in embedded devices, on such a scale, may lead to weaker protection.

e.g.
if using PSK, a single server compromise can put entire network at risk.
Devices performing critical tasks e.g. patient monitoring, medical implants etc. have even more associated risks.

MOTIVATION

2. **Resource constraint**: We cannot make direct switch from other authentication methods to PKI since modern PKI is not designed for constrained environments.

Hence, many embedded systems using low power lack the computing resources.

CHOOSING CERTIFICATES' FORMAT:

The paper follows X.509 certificates in its proposed PKI protocol.

REASONING

It was chosen to implement a public key infrastructure rather than any other authentication method since only PKI and digital certificates have the necessary mechanisms to provide strong authentication to devices of any size or function which are connected to the internet (claim).

Reinventing PKI for devices will take at least years for migration. They believe making constrained devices compatible with existing PKI is a better and more feasible approach (claim).

Moreover, none of the other approaches for implementing PKI in embedded systems were shown in practice. This paper presents a working model for the proposed protocol.

ASSUMPTIONS

1. NIST hash and cryptography recommendations are followed throughout this paper, regarding the algorithms and key lengths to be used.

2. Hardware of the constrained device, and the local software (running on the same device), are considered to be trusted.

ADDITIONAL WORK REQUIREMENTS

The proposed algorithm has no provisions of protection from possible DOS attacks. However, it is highly improbable for an attack to succeed in PKI4IoT (claim).

While the reasoning for this isn't explicitly mentioned in the paper, a possible reasoning is: An attacker who wants to take control of an IoT device would need to either compromise the CA, steal the private key of the device, or break the encryption algorithm, which are all very difficult tasks.

DISCUSSIONS ON RELATED WORK

1. Pre-shared key solutions (PSK) for IoT:
Security solutions based on PSK avoid higher computations of public key operations. However, there's a **trade-off between computations and security of the system**; the system is more vulnerable.

". A leaked shared secret phrase (used in PSK) requires updating all devices in the system, whereas a compromised certificate can be individually revoked"

DISCUSSIONS ON RELATED WORK

2. **Existing enrollment solutions**:

EST is found to be the best certificate enrollment standard for non-constrained devices.

(A possible reason to focus on non-constrained devices' enrollment standard is, as claimed by the paper, that it's better to modify existing standards rather than creating new ones, citing slow adoption as one of the issues.)

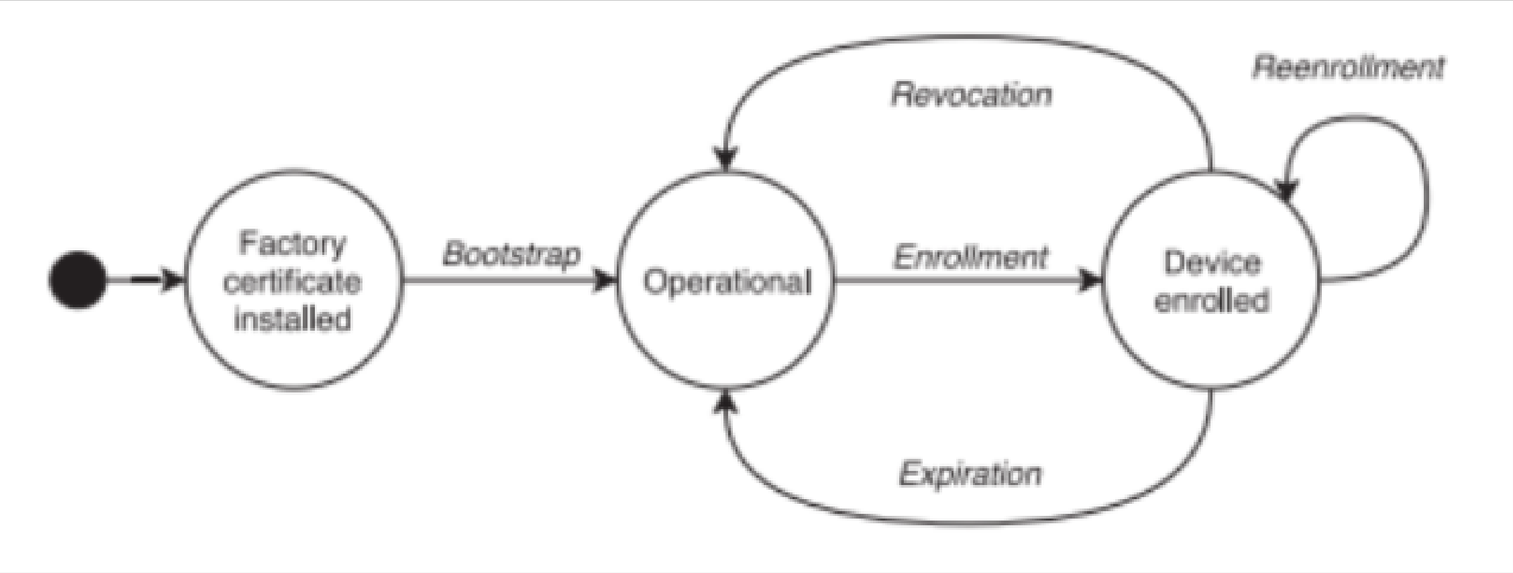
DISCUSSIONS ON RELATED WORK

3. **X.509 alternatives**

a. No security infrasture was found allowing implicit use of certificates when an IoT device wants to communicate with an arbitrary internet device.

b. None of the other attempts of creating ECC based certificates, which have similar size reductions (of the certificate), have been demonstrated in any PKI. This signifies the requirement of keeping X.509 compatibility.

LIFE CYCLE - PKI4IOT



1. BOOTSTRAPPING

1. Bootstrapping (“changing the state of a device/network/app from not operational to operational”)

The paper covers part of the bootstrapping page.
A pure standard-based solution encompassing all network layers (physical layer, data-link layer, network layer, transport layer, session layer etc.) is not defined since there isn’t a highly agreed-upon standard for bootstrapping.

2. CERTIFICATE ENROLLMENT AND RENEWAL

The document presents a solution for automated and secure certificate enrollment for IoT devices, based on EST over CoAPs. It also describes how to reduce the certificate usage overhead by using a lightweight profile and encoding for X.509 certificates, called XIOT.

3. CERTIFICATE RENOVATION

The paper suggests some alternatives to traditional certificate revocation procedures, such as OCSP, short validity periods, or CRLs, but notes that they have drawbacks for constrained devices and networks.
OCSP – less memory footprint, more communication steps
Traditional solutions – unacceptable overhead for constrained devices/networks

PKI4IOT : ADVANTAGES

1. **Communication overhead**: PKI4IoT reduces the size of X.509 certificates
by more than 50% using profiling and CBOR encoding. It also reduces the number of certificates needed for trust establishment by allowing the client to indicate that it only expects the server's own certificate.

2. **Energy consumption**: PKI4IoT reduces the energy expenditure for the client node by up to 58% compared to the case where a full certificate chain is sent. This is achieved by combining certificate compression and shortening techniques. The savings are more significant in multi-hop networks with lossy links.

3. **Round trip times**: PKI4IoT marginally improves the expected handshake time for the client node by reducing the number of bytes transmitted. The improvement is more noticeable in noisy radio environments, where packet losses and retransmissions are more frequent.

PKI4IOT : DISADVANTAGES

A lot of alternative methods to solve the problem of implementing PKI in embedded systems have been overlooked. The reasons of exclusions are mostly assertions; no strict arguments have been made for exclusion of alternative security solutions.

PKI4IOT : DISADVANTAGES

Presenting PKI as the only security solution for embedded systems (claim):

The authors claim PKI to be the only security solution that ensures maximum security in constrained devices/ or any device for that matter.

a. PKI still faces threat from quantum computers, capable of breaking classical crypto algorithms. This has been acknowledged by the authors, while also stating that such an issue is not a concern for the near future.

b. Other alternative security solutions exist, which could be implemented instead. Hardware security modules, multi-factor authentication, blockchain etc. to name a few. While these may require additional components in the constrained devices, even PKI4IoT requires comparable additional components.

Security & Efficiency

Several security measures for addressing the challenges of cloud storage:

- Data is encrypted throughout its entire life cycle using symmetric cipher encryption.
- One data one key encryption and decryption mechanism ensuring access to authorized users.
- Integrity of the data is protected through the use of a hash function.
- Data authentication is ensured through the use of digital signature ECDSA
- Security measures collectively work to safeguard the cloud data throughout its life cycle, ensuring data integrity, authentication, and access control.

New scheme for cloud storage security that addresses efficiency challenges:

- Uses symmetric cryptography algorithm for data encryption and decryption.
- Employs group secret as the parameter for Key Generation Function, making group sharing more efficient than other schemes using asymmetric cryptography for secret key distribution.
- Offers a more efficient solution to cloud storage security by leveraging symmetric cryptography for data encryption.
- It is group secret for key generation, reducing computation costs, and alleviating the burden of managing multiple secret keys for users.

Conclusion

Highlighting the increasing demand for secure cloud data storage services and the favorable characteristics of PKI-based cryptography, prompting the integration of these elements to address the security issues of data outsourcing in cloud storage systems.

Three models for cloud data storage are proposed, each accompanied by cryptography solutions to enable users to securely store and share data with authenticated users in the cloud.

ECC for all cryptographic operations is suggested, as it offers reduced computation and communication costs, along with smaller key sizes compared to RSA, while maintaining the same level of security, thereby enhancing scheme efficiency.

MULTI-LAYERED FRAMEWORK FOR SECURE SCADA COMMUNICATION:

SCADA stands for Supervisory Control and Data Acquisition. It refers to a system of software and hardware elements that allows organizations to control industrial processes, monitor and gather data in real-time from remote locations, and interact with devices such as valves, pumps, sensors, and more, typically found in industrial environments like manufacturing plants, power generation facilities, water treatment plants, and infrastructure systems.

MULTI-LAYERED FRAMEWORK FOR SECURE SCADA COMMUNICATION:

Key Elements of the Framework:

1. Key Generation: Session keys are created using a combination of random numbers, current date & time, and fractions of prime numbers. These elements are hashed to generate session keys, and their distribution is secured using MACSALT.

2. Key Distribution: The exchange of session keys, encrypted random numbers, and MACSALT is facilitated securely over the communication channel using asymmetric key cryptography.

3. Key Extraction: At the receiver end, the private and public keys are used to validate authentication and confidentiality, extracting the necessary elements to generate session keys

Secure Information Exchange:

• Control messages exchanged between devices are kept short and encrypted using a Vernam cipher, requiring distinct keys for each message.

• Different methods for symmetric key generation:

Hybrid Multi-Layered Architecture: Utilizes both symmetric and asymmetric key cryptography for secure communication.

Random Prime Number Generator: Generates session keys based on random primes for encryption and decryption.

Prime Counter: Utilizes a prime counter to enhance execution speed in generating session keys.

Hash Chaining: Provides high security and availability by using pre-shared keys and hash functions for key generation.

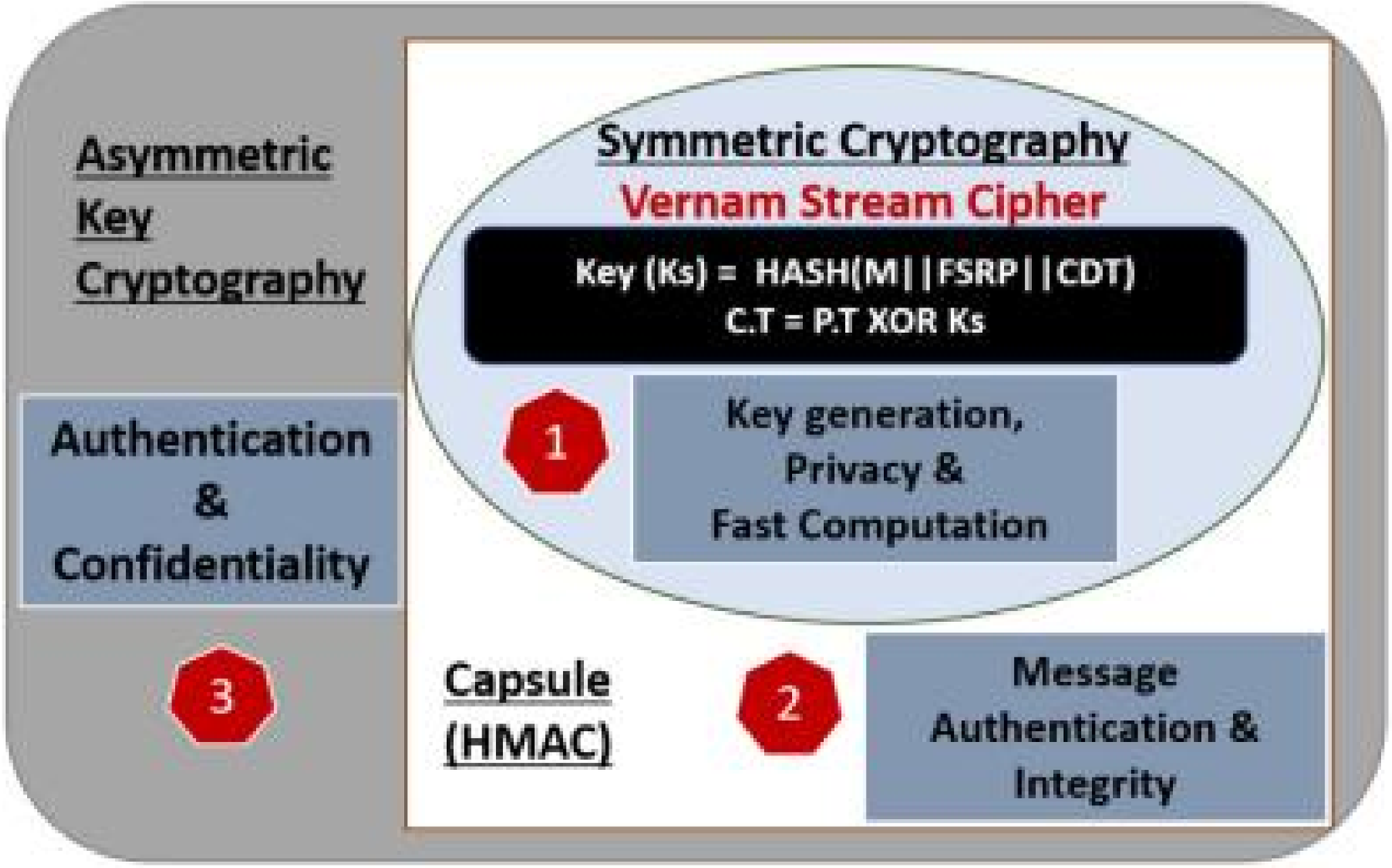


Fig. 2. Multi-layered framework for secure SCADA communication.

Conclusion

The proposed model intends to elevate security across various industrial sectors like water plants, power stations, chemical industries, and transportation systems. Successful implementation of this framework promises enhanced remote monitoring and control capabilities while fortifying the entire system against potential breaches.

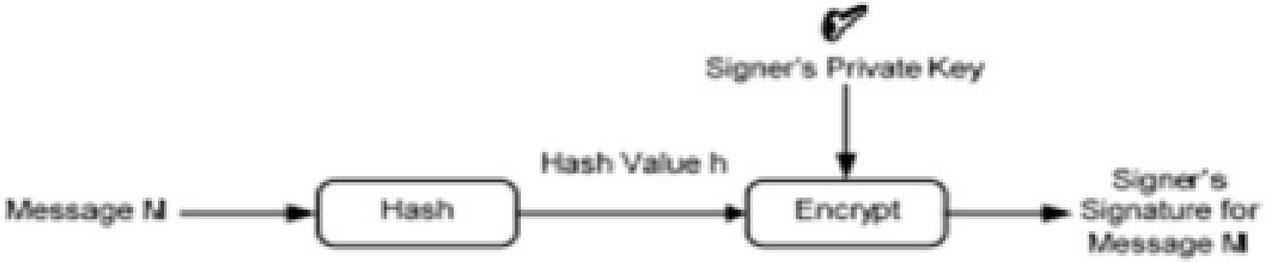
A Digital Signature Based on PKI To Authentication and Secure Exchanging data used in water boreholes intelligent Decision Support System

Introduction

Water scarcity is a crucial issue, particularly in regions like the Northern Sahara, where groundwater is a lifeline for drinking, irrigation, and industrial purposes. To achieve better exploitation, the authors suggest implementing a Decision Support System (DSS) to sustain the management of water boreholes in the southern part of the country. The data exchanged in their system is confidential and must be secured.

So, the authors introduced a digital signature system based on Public Key Infrastructure (PKI). The system is designed to secure the exchange of critical data within the DSS, ensuring the reliability and integrity of the information shared among users.

Proposed Approach *Based on a Digital Signature (DS)*



The **M** message is introduced to a cryptographic **h** function resulting in a **hash value h** (message digest). Using the private key, signer **encrypts** the h value (producing so the signature).

To check if the digital signature is valid, the h value of M is **compared** with the **signature's decryption value** using the public key (of the signer's).

If the two values are the **same**, the PK owner is the **message author**. **If not**, the signature is **not valid**.

Their approach for digitally signing the documents is described in two parts :

The signatory

- The value of the cryptographic hash of the data is calculated.
- This value is encrypted with its own private key (signature).
- The data, the signature, the identity of the signatory and the algorithms used are transmitted.

The receiver

- Receives D data and ES encrypted signature.
- Applies the cryptographic hash to D and finds H1.
- Retrieves the signer's public key from their identity.
- Uses it to decrypt the signature received ES and finds H2 – Compares H1 and H2 – The signature is valid if they are identical.

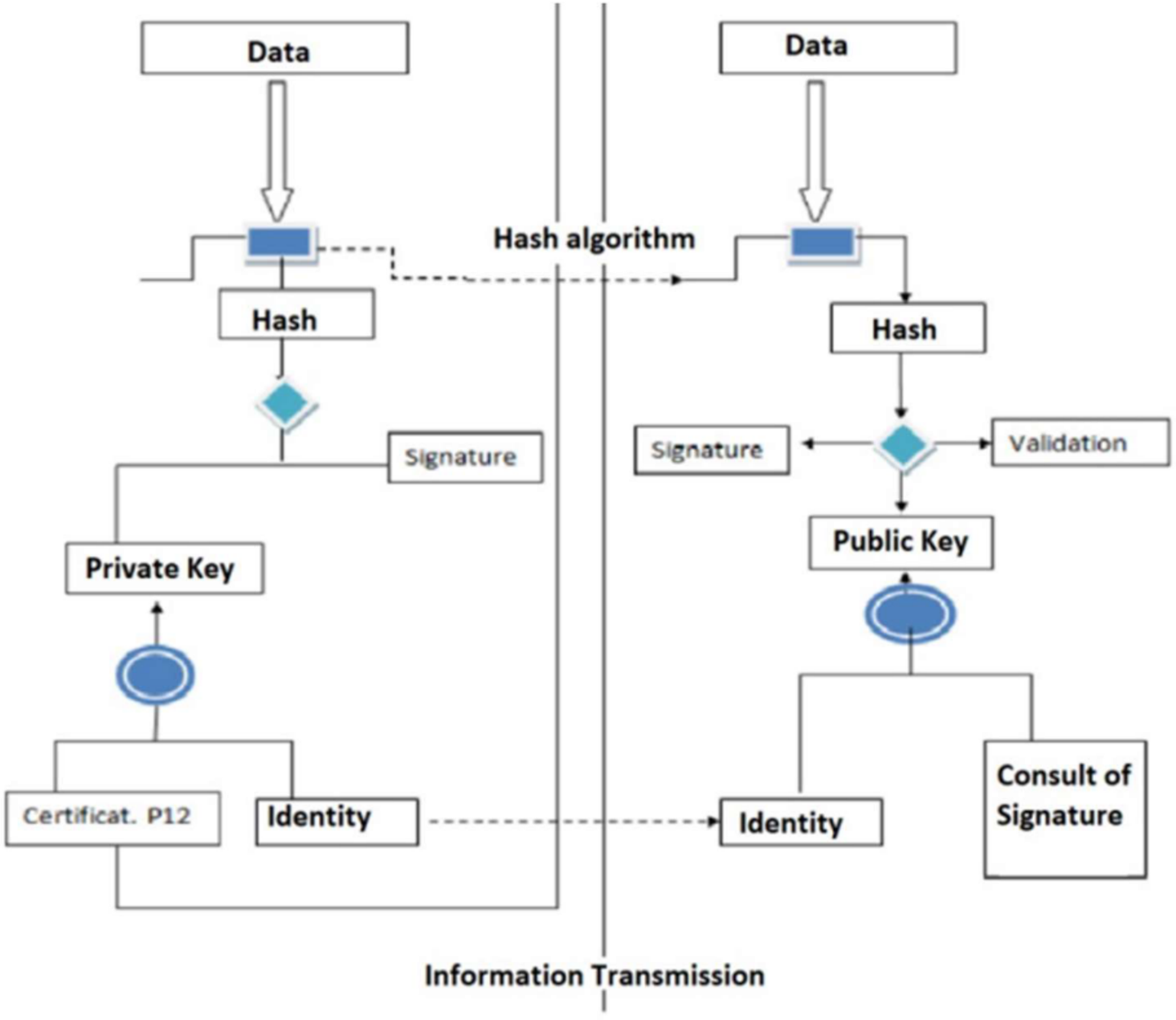


Fig. 3. Signature creation process.

Implementation of Proposed System

- To make it all work seamlessly, the team worked with **Java** language.
- For coding they chose **NetBeans IDE**.
- The team configured a **Linux server** using open **SSL** to serve as the backbone of their Public Key Infrastructure.
- For the **hash function**, they have used the **sha-256** algorithm to have the message digest of the data on our document.
- They used the **RSA** algorithm for the **encryption/decryption** with the private key/public key.
- They used **iText5**, an open-source library, to manipulate and create PDF documents.

ACTORS AND THEIR ROLES

Actor	Role
Administrator	Manages users and their certificates.
Signatory	Signs administrative documents using his private key.
Receiver or requestor	Signature requester. Verifies the signature when receiving the signed document.

Fig. 5. The certificate view.

Fig. 6. The document format Choose.

Fig. 7. Signing a PDF document interface.

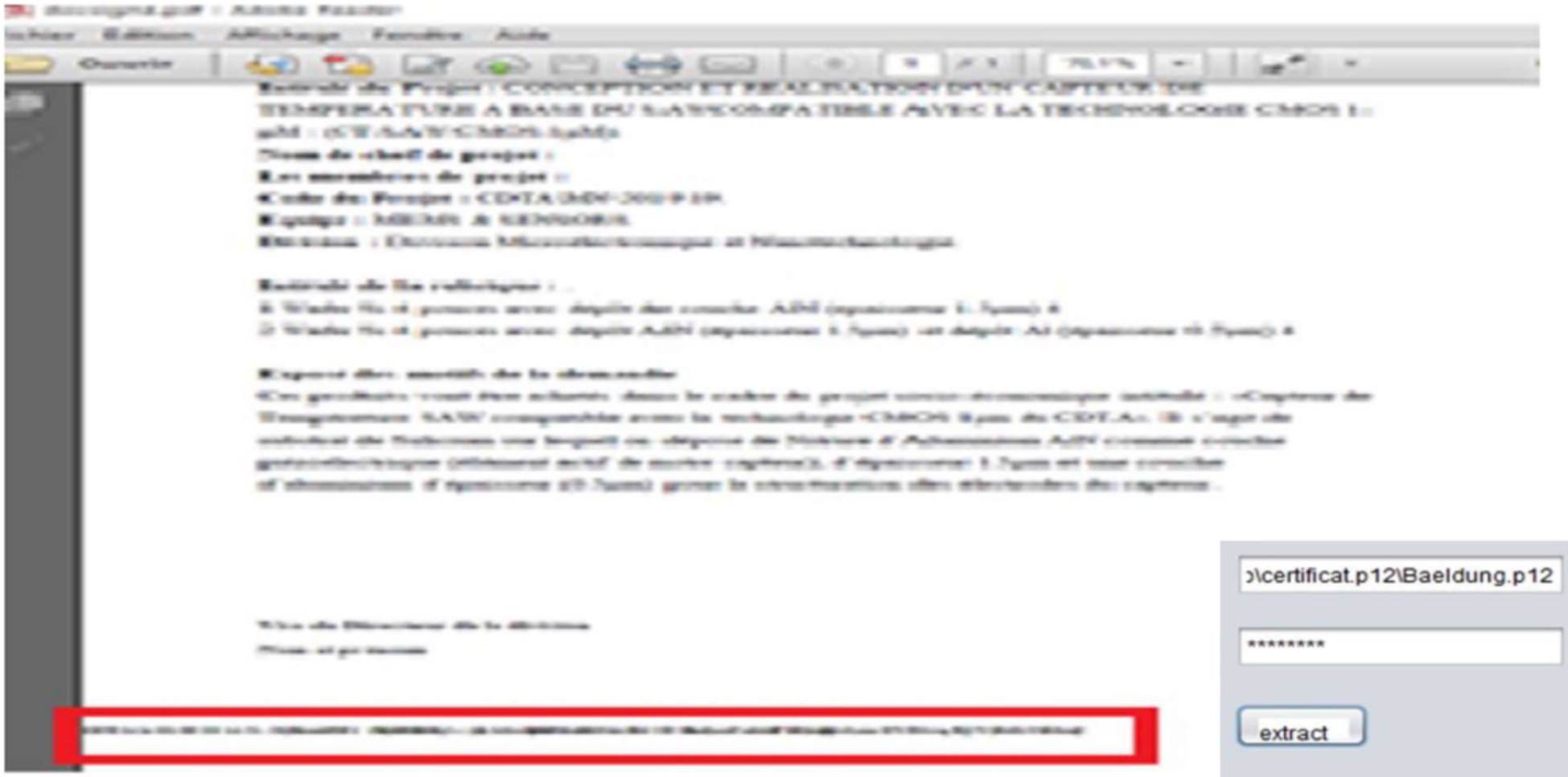


Fig. 8. The signed document.

Fig. 9. The public key extraction

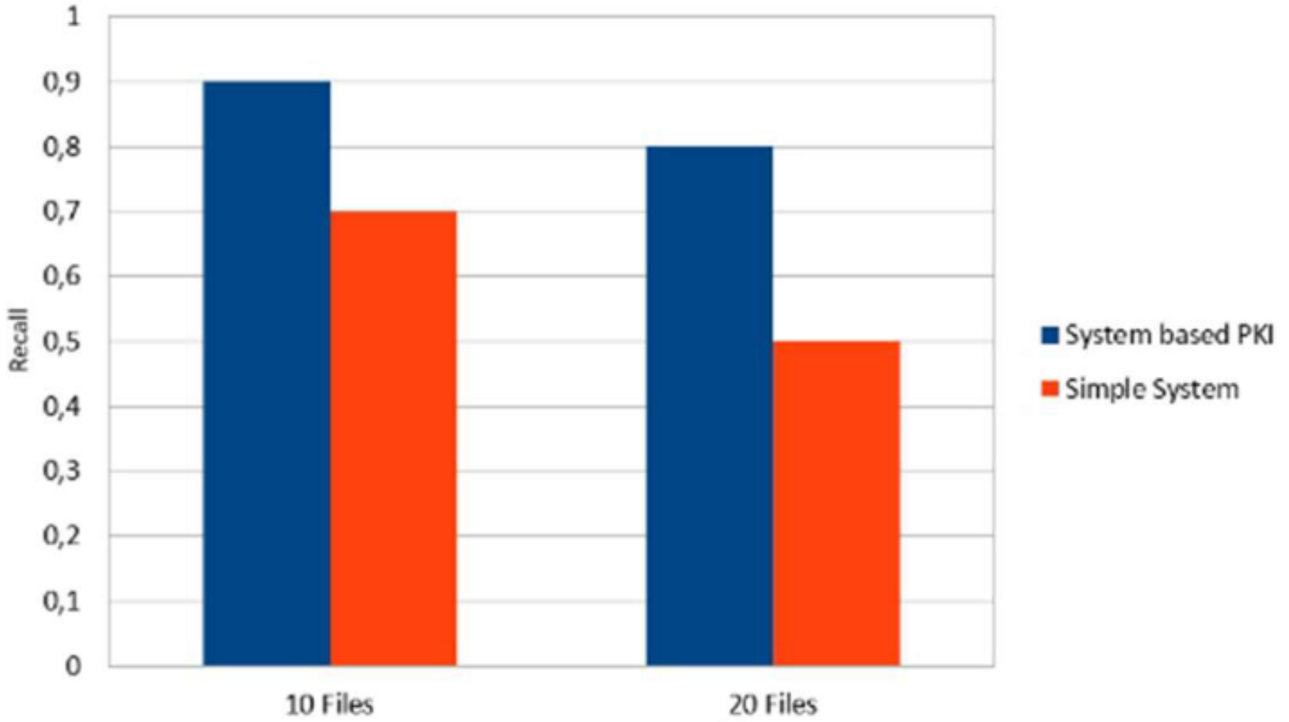
Fig. 10. The Signature verification.

Experimental Setup

- To test the approach, they made a comparison between a **system using the PKI signature** and another that uses **simple authentication methods**.
- To Evaluate, they used recall measure:

$$Recall = \frac{NumberOfAuthorizedAccessSignedKPI}{FilesNumberOfSignedPKIFiles}$$

- The goal here is that, the system should **only present relevant signed KPI documents to authorized users** during database queries.



- This model for secure information transfer consisted of following **quality aspects**:

  - **Safe Usage**
  - **Content Integrity**
  - **Confidentiality**
  - **Authentication of Sender & Receiver.**

References

K. Semar-Bitah, F. Z. Bouderbala and Y. Ghebghoub, "A Digital Signature Based on PKI To Authentication and Secure Exchanging data used in water boreholes intelligent Decision Support System," 2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAECCS), BLIDA, Algeria, 2023, pp. 1-5, doi: 10.1109/ICAECCS56710.2023.10105057.

PKI-Based Cryptography for Secure Cloud Data Storage Using ECC

Data safety in cloud storage and focus on the risks of unauthorized access and data tampering.
To solve the security issues, the paper looks at a few suggested solutions.

- using a type of encryption called attribute-based encryption
- electronic IDs for authentication
- a method called user hierarchy encryption
- a specific kind of cryptography called elliptic curve cryptography (ECC)

Elliptic Curve Cryptography (ECC)

It is a fundamental component of modern cryptography, initially proposed by Koblitz and Miller in 1985 in order to design public key cryptosystem and provide robust security for cloud storage solutions.
F_p denotes an elliptic curve E over a prime finite field F

$$y^2 = x^3 + ax + b$$

Where, $a, b \in F_p$ and the discriminate $D = 4a^3 + 27b^2 \neq 0$

$$A = \{(x, y) : x, y \in F_p, E(x, y) = 0\} \cup \{0\}$$

ECC operates within an additive group A, with a large order denoted by n, where G serves as the generator of A. Scalar point multiplication over A is defined, and the addition of points P and Q results in a point -R on the curve E/FP.

The security strength of ECC is attributed to the challenge of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP), and it provides equivalent security to RSA with a smaller key size.

Certificate Authority (CA) and PKI Enabled Application (PEA)

The Certification Authority (CA) is an important part of a system called Public Key Infrastructure (PKI). It's like a trusted agent that creates and gives out certificates.
When people or organizations want certificates, they share their details with the CA. The CA checks and confirms these details through a secure process.

Methadology

Data in the cloud is divided into private and shared parts

- private part is for storing sensitive data used only by the user
- shared part allows data to be shared with multiple authenticated users

Three cloud data storage models are designed based on this concept:

- Secure Data Backup model
- Secure Data Sharing One to One
- Secure Data Sharing One to Many.

Key generation function DKGen(S, Data id) is introduced, where S represents the user's secret information and Data id is the identifier of the data to be stored in the cloud. It generates different secret keys for different data, as symmetric cryptography is used for data encryption and decryption in the proposed cloud data storage models.

Secure data backup model for storing private and sensitive data in the cloud, utilizing attribute-based encryption.

Process begins with the data owner logging into the cloud infrastructure, sending a backup request to the cloud storage service (CSS), and receiving a data identifier in response.

Data is encrypted using a dynamically generated key and then uploaded to the private database, ensuring that each piece of data has a unique key and on retrieval, data is decrypted using the same key and its integrity is verified.

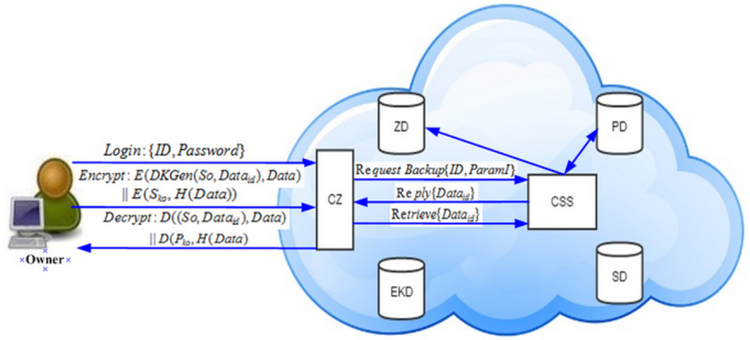


Fig. 1. Secure data backup mode

Secure data sharing one to one is for user to share data with other users in different group.

Data sharing one to one process,

User shares data via CSS:

1. Request: Owner ID, storage parameter 2 (for sharing), and user ID.
2. Response: CSS sends DataID and receiver's public key.
3. Upload: Owner generates secret key, hashes/signs data, encrypts key with receiver's public key, and sends to CSS.
4. Download: Receiver decrypts secret key, then data. Checks signature, compares hashes for integrity using owner's public key.

This ensures secure one-to-one data sharing, with the owner authorizing specific users to access their data in the cloud.

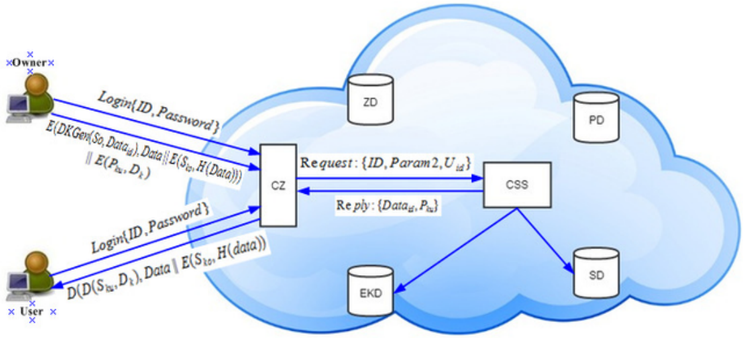


Fig. 2. Secure data sharing one to one

Secure data sharing one to many sharing is extended to one to many, group ID is used, secret function using Sg used for shared access and group secret key is used by group for access .

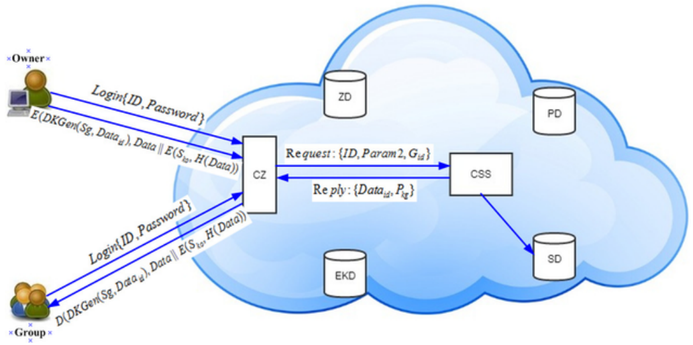


Fig. 3. Secure data sharing one to many

Secured User's Authentication and Private Data Storage- Access Scheme in Cloud Computing Using Elliptic Curve Cryptography

Elliptic Curve Encryption/Decryption:

Encryption Algorithm:

1. User A (Sender):

- Chooses a positive integer 'k' randomly.
- Has a private key 'd_A' and generates the public key 'P_A = d_A × G' (where 'G' is the base point on the elliptic curve).

2. Encryption Process:

- Message: 'P_m' (plaintext message)
- Public Key of B: 'P_B = d_B × G' (where 'd_B' is the private key of B)
- Encryption of Message:
- Calculate 'C_m = {kG, P_m + k × P_B}'
- Send 'C_m' as the encrypted message to User B.

Decryption Algorithm:

1. User B (Receiver):

- Has a private key 'd_B' and the public key 'P_B = d_B × G'.

2. Decryption Process:

- Received Cipher Text: 'C_m = {kG, P_m + k × P_B}'
- Decryption Steps:
- Calculate 'P_m = P_m + k × P_B - d_B × (kG)'
- Resulting 'P_m' is the decrypted original message.

Proposed Scheme:

1. Connection Establishment: The initial connection between user and cloud server is established with the help of HTTPS protocol, before the creation of account in the system.

2. Account Creation:

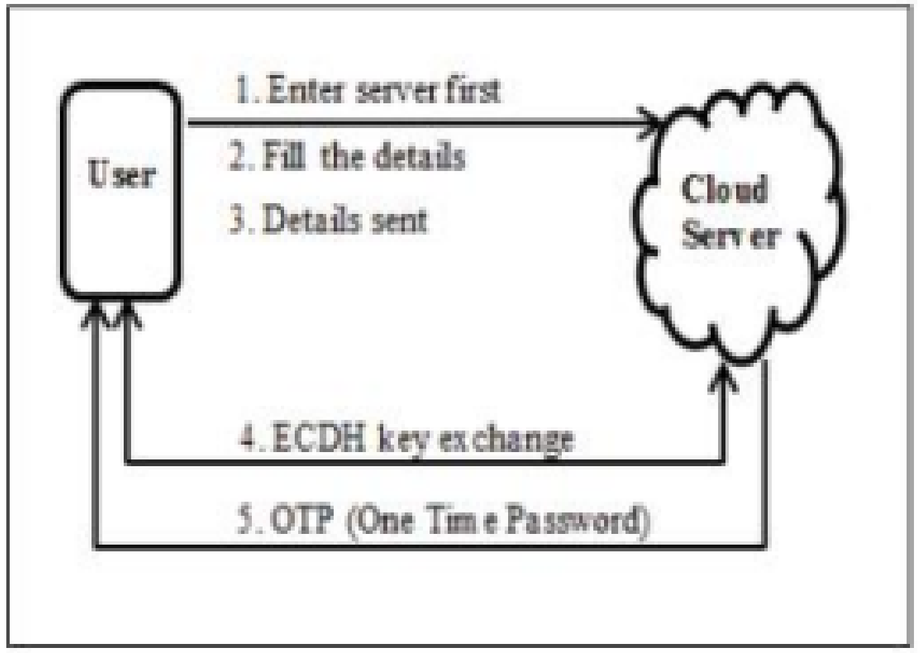


Fig.4.Account Creation

Proposed Scheme

3. User id generation:

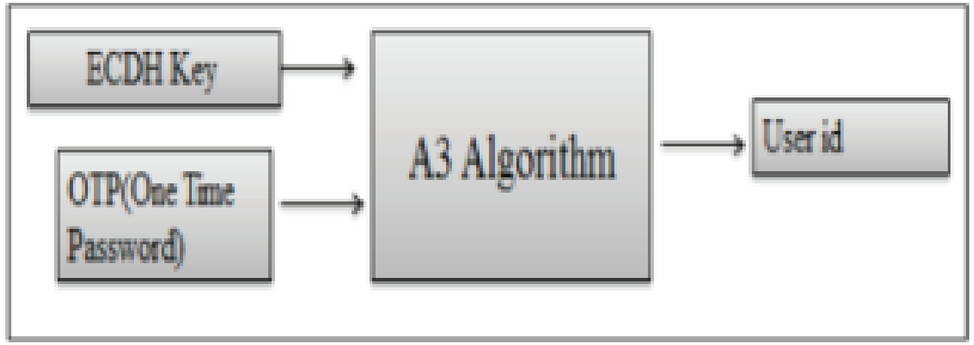


Fig.5. Input parameters for User id generation

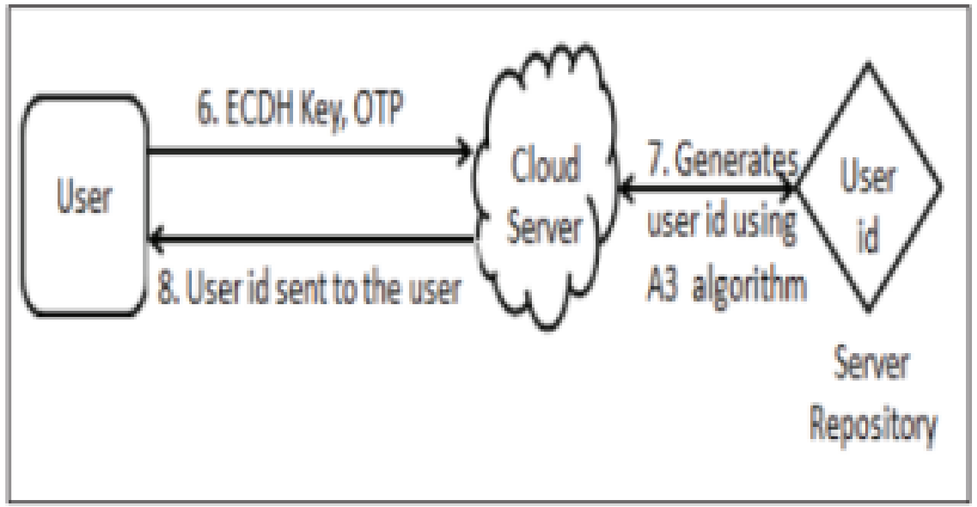


Fig.6.User id generation

Proposed Scheme

4. Authentication

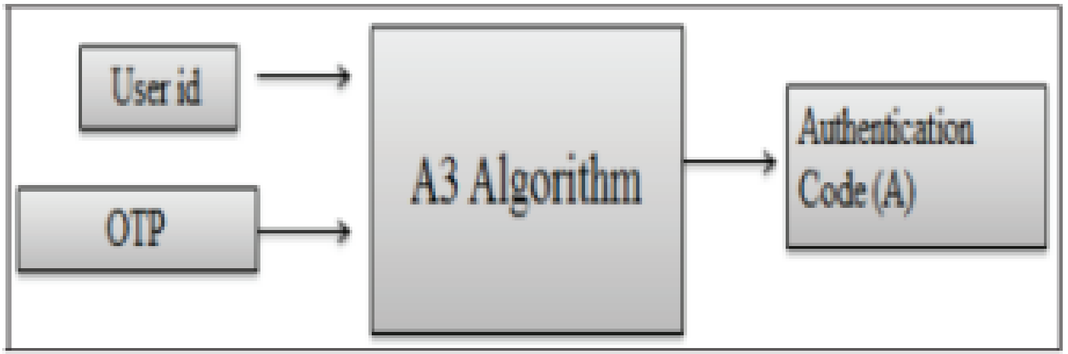


Fig.7. Authentication code generation at user end

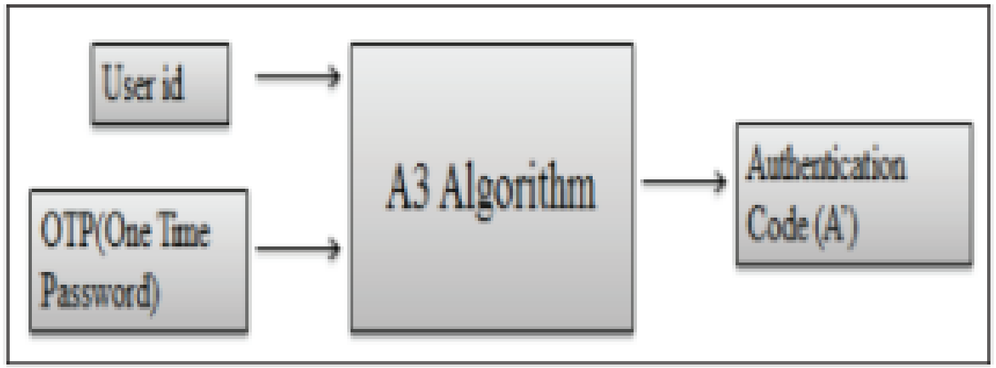


Fig.8. Authentication code generation at user end

Proposed Scheme

5. Encryption and Decryption: User first encrypts the data at the client side with the help of symmetric key algorithm and encrypts the key using ECC encryption algorithm and then uploads (encrypted data and encrypted key) it to the cloud. When the user needs the data, it must firstly decrypt the key using ECC decryption algorithm to download the data and then decrypts the data with his symmetric key.

A Survey of Key Bootstrapping Protocols Based on Public Key Cryptography in the Internet of Things

CLASSIFICATION APPROACHES OF KEY BOOTSTRAPPING PROTOCOLS:

1. Key Delivery Method:

Key Transport Mechanisms: Involve securely transferring a secret value from one party to another, often using out-of-band communication or pre-deployment. Includes one-pass, two-pass, and server-assisted approaches.

Key Agreement Mechanisms: Derive a shared secret from contributions of multiple parties, resisting eavesdropping. Utilizes algorithms like Diffie-Hellman and variants, ensuring higher security against attacks like man-in-the-middle.

2. Cryptographic Primitive:

Symmetric Key Pre-distribution Schemes: Parties share a common secret key for message encryption/decryption, ensuring implicit authentication. They rely on pre-established keys but suffer from scalability and vulnerability issues.

Asymmetric Key Schemes: Utilize public key cryptography (PKC) employing public-private key pairs for encryption, digital signatures, and authentication. Offer higher security but demand more computational resources.

CLASSIFICATION APPROACHES OF KEY BOOTSTRAPPING PROTOCOLS:

3. Authentication Mechanism:

- **Symmetric Cryptography-based Authentication:** Relies on pre-shared keys or established credentials, often using centralized servers for authentication.
- **Asymmetric Cryptography-based Authentication:** Involves peer-to-peer or managed methods. Peer-to-peer methods often utilize out-of-band channels for dynamic authentication, while managed methods rely on centralized servers.
- Other asymmetric authentication methods include identity-based authentication, PKI, certificate-based authentication, and cryptographically generated identifiers, each with its own security implications.

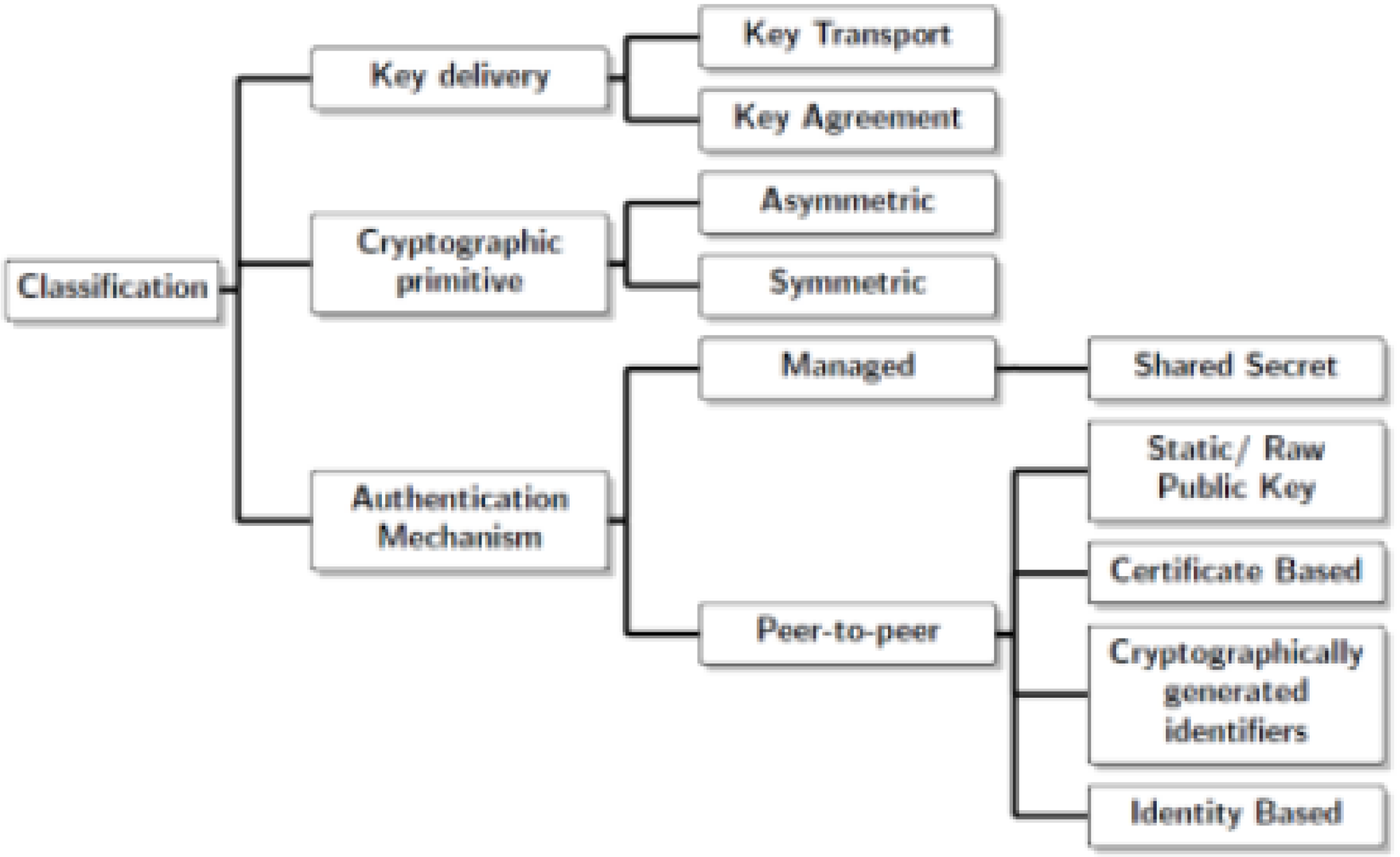


FIGURE 3. Classification approaches of key bootstrapping proposals.

ENPKESS

Encryption Process:

- Utilizes a non-linear Diophantine equation in the key generation phase to establish an enhanced and secure RSA variant.
- Implements a three-stage encryption process involving dual keys with associated polynomial equations, increasing key size for improved security.

Decryption Process:

- Employs a two-stage decryption process with dual keys, enhancing security while minimizing memory requirements.
- Integrates a packed knapsack mechanism, utilizing super-increasing integers and knapsack weights for encryption and decryption of matrix-form messages.

Packed Knapsack Encryption :

- The public keys α, R, N of the ENPKESS scheme are packed using the knapsack method. This involves transforming and arranging these keys based on a radix choice, resulting in a set $Ur = \{\alpha r, Rr, Nr\}$.
- The maximal element $max(l)$ is determined from this set, and a set L containing the lengths of elements in Ur is constructed. These elements are then used to create an unpacked matrix P, and its reversal matrix PV in row order is generated.

Thank you !

Nachiket(35)
Aryan15)

Harsh (20)
Ambedkar(11)

Rohan (47)
Pradeep(39)