Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 12, No. 1, March 2024, pp. 65~74
ISSN: 2089-3272, DOI: 10.52549/ijeei.v12i1.5392

A Privacy-Enhanced Scheme Within The Public Key Infrastructure For The Internet Of Things, Employing Elliptic Curve Diffie-Hellman (ECDH)

Abderrezzak Sebbah[1], Kadri Benamar[2]
[1] STIC Laboratory, Department of Computer Science, Faculty of Science, University of Abou Bekr Belkaid, Tlemcen, Algeria.
[2]STIC Laboratory, Department of Telecommunications, Faculty of Technology, University of Abou Bekr Belkaid, Tlemcen, Algeria

Article Info

Article history:

Received Jan 18, 2024
Revised Feb 4, 2024
Accepted Feb 18, 2024

Keywords:

IoT
Pseudonym
PKI
Authentication
Privacy
Security

ABSTRACT

The Public Key Infrastructure (PKI) serves as the foundation for online security, particularly within the realm of the Internet of Things (IoT). It operates based on certified public keys that remain permanent but can be revoked when necessary, such as in the case of a change in ownership, compromise of the private key, or malicious activities. Although this method ensures secure key utilization with traceability, it also introduces a potential privacy risk due to the traceability and utilization of identity-based certificates.

This approach is considered an innovative strategy for ensuring user confidentiality, integrity, authentication, and privacy in the context of the Internet of Things. The proposed solution integrates elliptic curves (ECDH) and traditional PKI to safeguard user privacy. It introduces two types of elliptic curve keys: long-term identity-based certified keys and dynamically generated temporary anonymous aliases. These aliases are seamlessly recorded by the certification authority, which maintains distinct directories for long-term and temporary keys. This dual-key approach enhances security while addressing the specific requirements of the Internet of Things.

Corresponding Author:

Sebbah Abderrezzak,
Department of Computer Science,
University of Abou Bekr Belkaid, Tlemcen, Algeria
Email: sebbahabderrezak1@gmail.com

1. INTRODUCTION

The Public Key Infrastructure (PKI) forms the foundation of secure services in the Internet of Things (IoT) domain, especially for services demanding authenticity, confidentiality, and integrity guarantees. PKI establishes the link between identity and the key through certificates. These certificates are created by a trusted Certification Authority (CA) and signed by it. This signature enables the verification of the certificate's integrity, current status, and authenticity [1].

PKI ensures traceability and accountability when using a certified key pair. The certificate can be updated to extend its validity or revoked by placing it in a revocation list. It contains crucial information related to the key owner, preventing any identity theft attempts, along with the validity period and public key.

Furthermore, it includes technical details such as the employed digital signature algorithm (DSA), hash function, public key, and the signer's identity. This data enables the recipient of the certificate to verify the validity, authenticity, and integrity of the associated public key. This method is known as "certificate-based authentication." Once this verification is completed, and the key is accepted, the owner assumes responsibility for all secure communications and applications in which it is used. However, it is important to note that the user's activity using this certified key is traceable. While traceability is a significant security feature, it poses privacy risks, especially in sensitive applications such as medical or financial records or industrial applications [1].

The field of industrial automation is one of the most crucial applications of the Internet of Things (IoT). By harnessing the capabilities of the Internet of Things, infrastructures incorporating advanced wireless sensor networks, and machine-to-machine communication are radically transforming conventional industrial automation and process control systems [2].

The Internet of Things is enabling industries to manage their operations more efficiently, simplifying tasks and eliminating errors. However, this evolution of the industrial Internet of Things is not without risks, particularly when it comes to confidentiality. The virtualization of physical objects and the massive collection of data raise privacy concerns. The sensitive data generated by industrial processes could be exposed, leading to potential risks of unauthorized disclosure or data compromise. It is, therefore, imperative for companies to put in place rigorous measures to guarantee the security and confidentiality of data in this intelligent environment.

In this study, we present a PKI system that combines ECDH (Elliptic Curve Diffie-Hellman) and uses two separate directories for long-term and short-term keys by the Certification Authority (CA), offering a more secure and efficient solution for managing certificates and aliases in an IoT environment. The use of ECC ensures secure communications while reducing key complexity, and the segmentation of directories allows for more precise management of long-term and short-term certificates. This approach enhances system security, confidentiality, and flexibility. This PKI is specifically tailored for privacy-sensitive industrial applications.

2. RELATED WORK

Public key cryptography relies on two distinct keys: a privately held key kept secret and a publicly accessible key available to all members of a community. This method ensures data confidentiality, integrity, and authentication of exchanged data. Key management, including their creation, distribution, renewal, and publication, is typically entrusted to a Certification Authority (CA), thereby forming a Public Key Infrastructure (PKI) [6].

The architecture of the Public Key Infrastructure (PKI) for the Internet of Things (IoT) aims to secure communications between various IoT devices. Specifically, PKI issues certificates to devices. However, the traditional PKI model relies on the need for Certification Authorities (trusted entities), such as the Internet X.509 Public Key Infrastructure [7], [8]. Unfortunately, this design has significant security and usability shortcomings, meaning that the entire structure is vulnerable if the Root-CA is attacked. Several recent security incidents highlight the vulnerability of CAs due to their centralized structure [9].

Several initiatives have been launched to explore alternative options, such as decentralized PKIs without Certification Authorities (CAs), which seek to eliminate the need for a trusted third party in the system. A pioneering example of this approach is the concept of the "Web of Trust," which enables the exchange of public keys without relying on a trusted entity (CA).

Thus, blockchain technology has garnered significant interest since its introduction in 2008. Several works have been proposed by [10], [11]. The challenge for lightweight clients, such as smartphone and IoT devices, lies in their limitations in terms of memory to store the entire blockchain. Currently, no solution has completely addressed this issue. Future research will focus on finding ways to enable these devices to operate normally while preserving their privacy.

3. THREAT MODELS CONCERNING PRIVACY IN IOT

Attacks targeting privacy in the Internet of Things (IoT) domain are a major concern, as they have the potential to compromise user data and the security of connected devices. Here is a list of common attacks that threaten privacy in IoT:

- Eavesdropping Attacks: An eavesdropping attack is a form of cyber-attack where attackers intrude into an ongoing conversation or data transmission, either by clandestinely listening or by pretending to be a legitimate participant. Attackers seek to intercept information exchanged between IoT devices, thereby exposing sensitive data. From the victim's perspective, the ongoing exchange appears to be normal communication, but by inserting themselves "in the middle" of the conversation or data transmission, the attacker can surreptitiously divert information [3].
- Traffic Analysis Attacks: Attackers analyze communication patterns between IoT devices, or the attacker intercepts messages exchanged between communicating parties, then carefully analyzes these messages to deduce their nature and content, as well as information about user activities [3].
- Identity Spoofing Attacks: Attackers impersonate legitimate IoT devices, enabling them to access confidential data or carry out attacks. A malicious node may propagate false routing information

to gain access to the confidential data of authentic nodes and thus impersonate their legitimacy within the network [4].
- Location Privacy Attacks: Malicious individuals strive to obtain the precise location of IoT devices or their users, which could potentially reveal sensitive or confidential information, thereby compromising the privacy of these individuals [1].
- Denial of Service (DoS) Attacks: DDoS attacks are carried out by multiple attackers in the network simultaneously. Some examples of DDoS attacks are flooding attacks that consume the bandwidth resources of the targeted system. Attackers inundate IoT devices with malicious traffic to render them unusable, endangering privacy [5].
- Data Correlation Attacks: Attackers aggregate data from various IoT sources to gain a more holistic perspective on a user's privacy [3].

4. SECURITY IN THE INTERNET OF THINGS

The Internet of Things (IoT) is employed in various sectors, including healthcare, industry, agriculture, and more. IoT devices primarily communicate via wireless technologies, which open the door to potential security threats to the data exchanged between these devices and users' smart-phones. Security challenges are particularly concerning for IoT protocol designers, as exemplified by a successful attack on medical devices in a hospital in 2015 [12]. To ensure the security and efficiency of IoT protocols, it is essential to consider the limited computational resources of these devices.

4.1 Security properties

To secure a mobile network, the following fundamental security objectives must be achieved:

- Confidentiality: Data must remain confidential and can only be understood by authorized parties, preventing unauthorized access, decryption, or interception.
- Authentication: This involves confirming the identity of a user, system, or entity to ensure their true declared identity, usually through evidence such as identifiers, passwords, or fingerprints.
- Integrity: Data must remain unchanged, uncorrupted, and unaltered during storage, transmission, or processing, ensuring its reliability.
- Non-Repudiation: This measure aims to prevent involved parties from denying their involvement or agreement in a transaction or agreement, which is essential for preventing future disputes.

4.2 Privacy

Preserving privacy is a crucial concern in the Internet of Things (IoT) domain due to the handling of sensitive personal data. This information pertains to user data related to this technology, including their actions, activities, habits, and interactions with other entities. Therefore, security and protection of this sensitive data become imperative, requiring the implementation of means to safeguard information across all IoT devices, user interfaces, and throughout the phases of data storage, communication, and processing. In order to protect the confidentiality and privacy of individuals connected to these devices, either directly or indirectly, the concept of privacy has been introduced. It can be summarized as follows:

- The user's right to have full control over information about them, with the ability to approve or deny its disclosure.
- The user's right to be certain that their data will only be used for the purposes for which it was provided.

Although the concept of privacy does not have a universal, uniform definition, its primary goal remains the protection of all user-related information, whether explicit or inferred, against disclosure, deduction, knowledge, or use without the explicit consent of the user. For example:

- *Electronic Payment:* Information provided by a user for electronic payment should not be used to track them, monitor their activities, or discover their purchasing habits without their explicit consent.
- *Location and Traceability:* When a user grants an IoT application access to their geographic location for specific purposes, it is essential to emphasize that this permission should never be used to track or target advertising without their explicit consent.
- *Health and Medical Data:* Medical information gathered by IoT devices, such as heart monitors or glucometers, is highly sensitive. In this context, it is essential for the user to maintain absolute control over who can access this data and for what purposes, whether it be healthcare professionals, researchers, or medical monitoring applications.

- *Smart Home and Security Cameras:* In the context of a smart home, IoT security cameras are designed to secure the home. However, it is essential that they are not used to monitor the daily lives of residents without their permission.
- *Connected Cars:* Connected cars collect a vast amount of data, including driving habits and geographic location. The user's control over this information must be absolute.

In the context of the Internet of Things (IoT), considerable efforts have been made to address privacy issues [13], [14], [15]. Some of these approaches rely on cryptographic techniques, such as anonymity or pseudonymity (Alias), while others focus more on establishing policies or rules to ensure privacy through a negotiation process between the Client/Server (information provider/information consumer). The aim of this exchange is to determine the minimum amount of information to disclose to access the desired service.

5. SECURITY MECHANISMS

5.1 Public Key Infrastructure (PKI)

A Key Management Infrastructure, commonly known as Public Key Infrastructure (PKI), refers to an integrated set of technologies, processes, and software designed to ensure the secure management of digital certificates throughout their lifecycle. Digital certificates, or electronic certificates, are crucial for securing electronic transactions [16]. They facilitate cryptographic operations such as encryption and digital signatures, providing essential assurances: Confidentiality, Authentication, Integrity, Non-Repudiation, and more.

To achieve these objectives, the management of the PKI infrastructure relies on several essential entities integrated within the PKI system. These entities include the Certification Authority (CA), Registration Authority (RA), Repository Authority, and End Entity (EE). Each of these entities serves a specific function in overseeing certificates and keys [15].

The Certification Authority (CA) is a central trust entity that plays a crucial role within a Public Key Infrastructure (PKI). Its primary role encompasses certificate generation, validation, and digital signing. Additionally, it determines certification policies and establishes certification practice statements.

- The Registration Authority (RA) is responsible for verifying the identity of individuals, entities, or computing devices before submitting a certificate request to the CA.
- The Repository Authority serves as a centralized repository where users can access and retrieve public certificates of other trusted entities. Furthermore, the Repository Authority regularly publishes Certificate Revocation Lists (CRLs), listing revoked certificates. This allows users to verify the validity of circulating certificates.
- The End Entity (EE) uses this certificate to perform various cryptographic operations, such as data encryption or digital signatures.

PKI finds extensive utility in various application scenarios, including:

- Using SSL certificates to secure websites and publicly accessible services.
- Implementing PKI in private networks and VPNs to ensure secure communications.
- Integrating digital certificates into applications and services hosted on public cloud platforms, ensuring data protection.

However, this PKI approach has drawbacks related to the need for certification authorities, which pose security, usability, and privacy challenges, particularly in maintaining user traceability, which can lead to privacy risks.

5.2 Elliptic Curve Cryptography (ECC)

The use of elliptic curves in cryptography was independently proposed in 1985 by [17], [18]. Elliptic curves used in this context are defined by a simplified Weierstrass affine equation:

$$E: y^2 = x^3 + ax + b \ (*)$$

These elliptic curves can be represented in different sets of numbers. In cryptography, they are generally represented in finite fields such as Fp (where p is a prime number) and F2n. The coefficients a, b uniquely identify an elliptic curve. The solutions to equation (*) are points Pi (xi, yi) belonging to the elliptic curve.

Adding two points: Adding two points on the elliptic curve results in a third point that also resides on the curve.

Scalar multiplication can be defined by applying the point addition operation k times. This operation is relatively simple to perform with integers, but the inverse operation, carried out using discrete logarithms on integers, is computationally challenging. This is why elliptic curve cryptography relies on the Discrete Logarithm Problem (DLP) to ensure its security.

For example, the operation: S = k * P (**), represents the addition of point P to itself k times.

For example, if k=3, the operation (**) becomes Q = 3 * P, which is equivalent to (2 * P) + P, equivalent to P + P + P [19].

Elliptic curves offer increased security relative to the key size used while being particularly resistant to attacks. Unlike conventional cryptography methods like RSA, where keys are generated from complex operations, ECC keys are derived from a line drawn on an elliptic curve. Therefore, a 256-bit ECC key can offer a level of security equivalent to a 3072-bit RSA key. This approach has the advantage of reducing the required storage space and bandwidth when transmitting data [19].

5.3 Elliptic curve diffie-hellman (ECDH)

Elliptic Curve Diffie-Hellman (ECDH) is a variant of the Diffie-Hellman algorithm based on elliptic curves. It closely resembles the traditional Diffie-Hellman algorithm used for secure key exchange. Unlike Diffie-Hellman, which involves exponential operations on keys, ECDH uses elliptic curve cryptography for multiplication [20], making it a relatively simple protocol. Here's an example: Node **A** and Node **B** choose an elliptic curve, a prime number P, and a point G (Generator) on the curve.

- ✓ Node **A** selects a random secret number z-a.
- ✓ Node **A** calculates point **A** = z-a * G and sends it to Node **B**.
- ✓ Node **B** selects a random secret number z-b.
- ✓ Node **B** calculates point **B** = z-b * G and sends it to Node **A**.
- ✓ Node **A** computes the shared secret as xk = z-a * **B** = z-a * z-b * G, and Node **B** computes it as xk = z-b * **A** = z-b * z-a * G.

The shared secret calculated by both parties is identical because z-a * **B** = z-a * z-b * G = z-b * **A**.

6. MOTIVATION

The quality of a security protocol depends on various factors, such as the type of security mechanism used, the network structure, and the specific communication context. To assess the security effectiveness of a protocol, it is essential to consider the inherent specifics of the network in which it will be implemented.

- The conventional Public Key Infrastructure (PKI) based on the Certificate Authority (CA) has various vulnerabilities. Who do we trust, and for what? The question is whether the information provided to the CA is sufficient to trust the user and certify their keys. The CA is an authority in creating certificates but not elsewhere, so it raises the question: Is the CA an authority?
- Who is using my key? Non-repudiation holds the key owner responsible for activities performed with that key.
- These vulnerabilities can result from weak design or implementation of the PKI. One of the main issues with the current PKI is the single point of failure problem that can target the CA and disrupt its availability. These vulnerabilities lead to the question: How secure are certification practices

6.1 Assumption

In this paper, we introduce an innovative proposal for Internt of Things that relies on the use of elliptic curves (ECDH) and traditional PKI to ensure user privacy. The main contributions of this work are as follows:

1) Our system is based on anonymous certificates, the aliases generated using ECDH encryption.
2) The proposed PKI is based on two separate directories for certificates and generated aliases. The first directory contains certificates (RCL), while the second directory is reserved for aliases and contains a revocation list (RRCT) that is archived.
3) Our system is secure against various attacks.

6.2 Network Model

Figure 1 illustrates the flow of communication between different elements of the Internet of Things (IoT), emphasizing security and privacy protection. This illustration provides a visual summary of secure communications within the IoT framework, highlighting the importance of security and privacy protection. It explains how IoT devices can exchange information with confidence without compromising user privacy.

- **IoT Devices:** At the core of the figure, several IoT devices are represented as icons, including various devices such as sensors, home appliances, connected vehicles, etc. These devices interact with each other to exchange data.
- **IoT Gateway:** Positioned at the periphery, IoT gateways are present to facilitate communication between IoT devices and the external network. Gateways serve as secure access points.
- **Certificate Authority (CA)**: At the top of the diagram, an icon represents the Certificate Authority (CA). The Certificate Authority is the central point of security in the IoT ecosystem.
- **Communication Flow**: Bi-directional arrows connect IoT devices to IoT gateways, and then to servers or the external Internet. These arrows symbolize secure data exchanges.

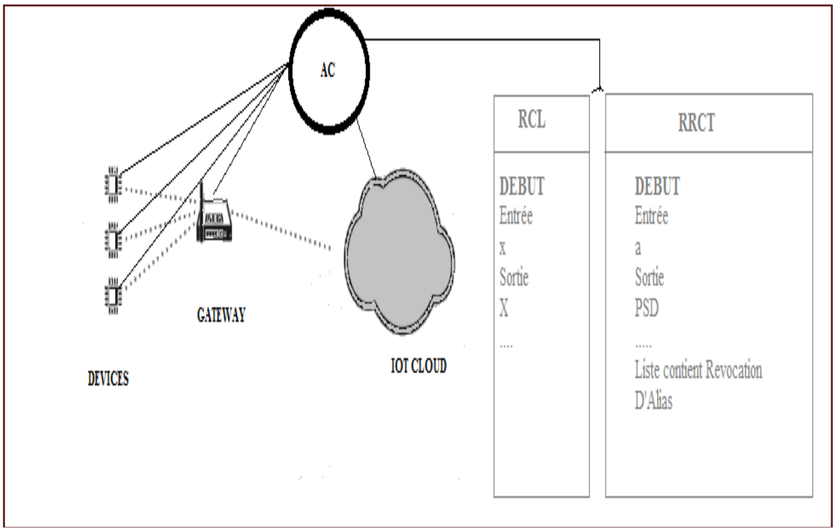

Figure 1. The flow of communication between different elements of the IoT

7. THE PROPOSED SOLUTION

In this section, we present our protocol that relies on Elliptic Curve Cryptography (ECC) and Public Key Infrastructure (PKI) to establish an association between identity and the key using certificates. These certificates are generated by a trusted Certificate Authority (CA) and validated by it. This validation ensures the integrity, freshness, and authenticity of the certificates.

The Certificate Authority maintains two separate directories for long-term keys and temporary keys. The first directory contains certificates (RCL), while the second directory is reserved for aliases and contains the Revocation List (RRCT), which is used for the certification of short-term keys, also known as aliases. Both directories are archived by the Certificate Authority, ensuring secure and traceable key management in the certification process.

The suggested network model consists of these phases: pre-deployment, alias generation, alias revocation, and alias verification and authentication with IoT devices. These phases enable authentication between the user and the Certificate Authority (Algorithm 1 summarizes these steps)

Step 1: Pre-deployment

An IoT device must establish secure communication with an authorized user. The IoT device generates a pair of ECDH keys:

Device IoT: private key (dev)

Device IoT: public key (DEV = dev * P)

Initial Registration: When a user wishes to register with the Certificate Authority (CA), they provide their identity information. This information, assumed to be authentic and sufficient for user identification, is used by the CA to issue a certificate to the user.

Long-term Certificate: The long-term certificate includes two pairs of secret/public keys, denoted as (x, X), where $x \in Z^*n$, and $X = x * P$. These keys are used to ensure communication security.

Step 2: Alias Generation

Alias Request: When the user needs an alias (pseudonym), they securely send a request to the CA. This request must include the user's long-term certificate containing the public key X.

Revocation Check: The CA starts by verifying if the requesting user has not been revoked, i.e., if X is not in the Certificate Revocation List (CRL) maintained by the CA.
Alias Generation: If the user is authorized to obtain a pseudonym, the CA generates a random number a.
Alias Calculation: The CA then generates the alias PSD for the user using a and X as follows:
$PSD = a * X$
Note: PSD is a public key. Additionally, only the user can calculate the corresponding private key.
Alias Publication: The CA publishes the generated PSD, its validity period, and the random number a in a directory reserved for temporary keys (RRCT). This publication allows the user to look up their requested alias in this directory (RRCT).
Secure Backup: The CA securely backs up the generated PSD and the certificate containing key X. This ensures traceability and revocation in case of malicious behavior.
Obtaining the Corresponding Private Key: To obtain the private key (x-z) corresponding to the generated PSD, the user performs the following calculation:
$PSD = a * X$
The shared secret calculated by both parties is identical because:
$a * X = a * x * P = a * X$
In fact, only the user can derive the confidential key x-z from the published PSD and a since they are the only ones who know the value of x.

Step 3: Alias Revocation by the Certificate Authority (CA)
Alias Revocation: To revoke an alias, the Certificate Authority (CA) uses the PSD received in the report of malicious behavior.
Searching for Revoked PSD: The CA searches for the reported PSD to find its match with X, to facilitate the search and make it efficient, the CA relies on the validity period to search only within the blocks created during the PSD's validity period.
Insertion into the Certificate Revocation List (CRL): When the CA finds X, it inserts their corresponding certificate into its Certificate Revocation List (CRL). This action prevents the user from requesting further aliases in the future.
CRL Sharing: The CRL is shared among the (CA) that form and maintain the RRCT. It is exchanged securely between them.

Step 4: Alias Verification
Before accepting an alias, the receiving node must perform the following checks:

- ✓ Verify the PSD in the directory reserved for temporary keys (RRCT).
- ✓ Verify the validity period of the PSD.
- ✓ Verify the PSD in the Certificate Revocation List (CRL).

Step 5: Authentication with IoT Devices
ECDH Key Exchange: The IoT device intends to transmit data securely to the user. In this process, it acquires the user's certificate (PSD), which contains the user's public key. The IoT device implements the ECDH method to create a shared session key (K-S) using the user's public key:
$K\text{-}S = ECDH\ (dev, PSD) = dev * PSD = dev * a * X = dev * (a * x) = dev * x\text{-}z * P$
Secure Communication: The IoT device encrypts the data to be sent to the user with the session key K-S and sends the encrypted data to the user.
Reception and Decryption: The legitimate user receives the encrypted data from the IoT device. The user uses the ECDH method to create an identical shared session key (K-S). This key is generated from their own private key (x-z) and the IoT device's public key (DEV) included in the IoT device's certificate:

$$K\text{-}S = ECDH\ (x\text{-}z, DEV) = x\text{-}z * dev * P = x\text{-}z * dev * P.$$

# Algorithm 1: Alias Process

```
# Alias Generation
Function generate_Alias (User):
  If User is not revoked:
    a = generateRandomNumber()
    PSD = a * User.publicKey(X)
    PublishIn (RRCT) (PSD, validity, a)
    SavePSDandCertificate (PSD, User.publicKey)
    Return PSD
  End
```

```
# Alias Revocation
Function revoke_Alias (Reported_PSD):
  X = find_X_from_PSD (Reported_PSD, validityPeriod)
  Add X to CRL(X)

# Alias Verification
Function verify_Alias (PSD):
  If PSD exists in (RRCT):
    If validityPeriod is valid:
      If PSD not present in the revocation list:
        AcceptAlias ()
        "Authentication successfully completed"
      Else:
        RejectAlias ()
        "Authentication failed"
      End If.
    "Authentication failed"
  End If.
"Authentication failed"
End.
```

8. SECURITY ANALYSIS

This section conducts a thorough evaluation of the security of the proposed protocol design based on various types of attacks on emerging IoT devices. The proposed solution ensures several security features, including:

- **Privacy Concept:** Each component of the connection request dynamically varies using a random number 'a.' Therefore, when an attacker intercepts connection requests, it becomes impossible to trace a specific user, ensuring anonymity preservation. Our approach guarantees complete confidentiality. The temporary keys (aliases) stored in the Directory lack identification information (anonymous). The user's personal data included in the long-term certificate containing 'X' is securely exchanged with the Certification Authority during alias request.
- **Mutual Authentication**: Each step of the proposed system has specific authorization based on a shared secret key during the Alias verification phase. User Ui and the CA mutually authenticate and establish a private session key using PSD, a, while verifying the following:
  $PSD = a * X$.
  The shared secret computed by both parties is identical because:
  $a * X = a * x * P = a * X$.
  The proposed protocol ensures mutual authentication.
- **Integrity:** Aliases are generated by the Certification Authority (CA) using a random number and the owner's public key. Once a pseudonym is generated, it is logged in the directories, ensuring its integrity. Any attempt to modify or alter the pseudonym would be immediately detected. Therefore, Aliases benefit from integrity through their public and immutable recording.
- **Non-Repudiation:** In the context of Aliases, once a pseudonym is registered, its owner cannot repudiate or deny being its creator or holder. This aligns perfectly with the principle of data integrity stored in the directory.
- **Freshness:** Alias transactions found in the Certification Authority's Directories have timestamps and specific validity periods thanks to the random number a. Our approach is also capable of resisting the following types of attacks:
- **Replay Attacks:** The protocol ensures its security through the use of a random numbering technique. Each user Ui and the Certification Authority (CA) generate random values x and a. Incorporating these random values ensures that extended messages remain constantly up-to-date and fresh, preventing replay attacks.
- **Distributed Denial of Service (DDoS) Attack:** To attack the availability of directory-based PKI and render it paralyzed, all nodes must be attacked simultaneously. To illustrate the idea that all nodes must be attacked simultaneously, consider a simplified example: Imagine a group of people

who rely on a shared virtual library in a cloud computing network to access their e-books. Each person can download these books from any network node to read them.

In this scenario: Network cloud nodes represent the servers where the virtual library is stored. Users of the network are the individuals who want to read the e-books. Now, consider that a malicious group wants to disrupt access to the virtual library. To do so, they decide to launch a Distributed Denial of Service (DDoS) attack. Now, if the malicious group coordinates their attack in such a way that all members attack all cloud servers simultaneously, then all servers become inaccessible at the same time. Users can no longer access their e-books because the entire service is paralyzed. This is equivalent to what would be required to paralyze directory-based PKI.

- **Forward Secrecy:** Suppose an attacker attempts to compromise the long-term parameters of active entities, including private and public keys of a tag and a reader. To create a session key, we use PSD and X. These parameters (PSD and X) are generated using two random numbers, x and a. Even if an attacker manages to obtain the private key of an entity, they remain unable to calculate the session key due to ECDHP theorems. Consequently, they cannot generate the session key, ensuring security against forward secrecy attacks.
- **Stolen Verify Attacks:** Imagine an attacker attempting a Stolen Verify Attack by seeking access to the long-term parameters of active entities, such as private and public keys. To generate a session key, we use PSD = a * X (Pseudo-Shared Secret) and X = x * P.

  These parameters (PSD and X) are created using two random numbers, x and a. Elliptic curve Diffie-Hellman theorems ensure that even if the attacker obtains the public key of an entity, which is X in our case, and the private key of another entity, which could be the tag or reader's private key, they cannot calculate the session key. This means that the attacker cannot interact with the system fraudulently because they cannot generate the necessary session key for secure communication.
- **Impersonation:** Suppose an attacker records one of these requests for later use. In this case, the Certification Authority generates temporary aliases using the certificate's public key. Therefore, only the legitimate owners of these aliases possess the corresponding private keys, making it impossible for the imposter to decrypt data encrypted with these aliases or use them to sign messages, as the imposter does not have access to this sensitive information. Let's take an example where an attacker A wishes to impersonate user U and request aliases on behalf of U to avoid being traceable. A sends a secure alias request containing Certificate U. The Certification Authorities generate alias PSD and insert it into the directory alongside a-new.

  PSD = X * a-new. A retrieves PSD from the directory, but to use it, A must be able to calculate x-z, the pair of private keys for PSD, x-z = x * a-new. Since x is the private key known only to user U, attacker A cannot impersonate U.

9. CONCLUSION

This article presents an innovative proposal for a Public Key Infrastructure (PKI) based on two directories, RRCT/RCL. Our approach skillfully combines the principles of a traditional PKI with ECDH to ensure user confidentiality. Our system relies on anonymous certificates, which we refer to as "Aliases" or pseudonyms. These pseudonyms are generated from the original public key using the ECC (Elliptic Curve Cryptography) algorithm, thereby establishing a link between users' real identities and their pseudonyms. One of the key features of our approach is that no legitimate authority can reveal a user's real identity from their pseudonym, ensuring a high level of privacy.

To enhance the security and resilience of our approach, we have taken measures to minimize vulnerabilities related to a single point of failure and various popular attacks. This robust and resilient design is based on a lightweight architecture, making it suitable for a wide range of applications.

REFERENCES

[1] Leila Benarous and Benamar Kadri, “ The quest of privacy in public key infrastructure,” *International Journal of Blockchains and Cryptocurrencies*, Vol. 2, No. 3, 2021.

[2] Zahera DIB, imadeddine RAHMOUNI, Souici KHOUDIR, Walid DIB, “ Application of the internet of things in the traceability of a supply chain,” *Academic Journal of Manufacturing Engineering*, 2023.

[3] S. Pundir, M. Wazid, D. P. Singh, A. K. Das, J. J. P. C. Rodrigues and Y. Park, "Intrusion Detection Protocols in Wireless Sensor Networks Integrated to Internet of Things Deployment: Survey and Future Challenges," *in IEEE Access*, vol. 8, pp. 3343-3363, 2020, doi: 10.1109/ACCESS.2019.2962829.

[4] Ahmed, A.A.; Malebary, S.J.; Ali, W.; Alzahrani, A.A, “A Provable Secure Cybersecurity Mechanism Based on Combination of Lightweight Cryptography and Authentication for Internet of Things,” *Mathematics*, 2023, 11, 220. https://doi.org/10.3390/math11010220

[5] Ahmed, A.A.; Ahmed, W.A, “An Effective Multifactor Authentication Mechanism Based on Combiners of Hash Function over Internet of Things,” *Sensors*, 2019, 19, 366

[6] Kadri, Benamar & Feham, Mohammed & M'Hamed, Abdallah, “Lightweight PKI for WSNs uPKI,” *International Journal of Network Security*. 10, 2010.

[7] C. Adams and S. Farrell, “ RFC2510: Internet X.509 Public Key Infrastructure Certificate Management Protocols,” *RFC Editor*, USA, 1999.

[8] EllisonC. et al, “Ten risks of PKI: What you’re not being told about public key infrastructure,” *Comput. Secur. J.* (2000).

[9] Zhu J, Wan C, Nie P, Chen Y, Su Z, “Guided, Deep Testing of X. 509 Certificate Validation via Coverage Transfer Graphs*,” In: IEEE. ;* 2020: 243–254.

[10] Lei, A., Cruickshank, H.S., Cao, Y., Asuquo, P.M., Ogah, C.P., & Sun, Z, “ Blockchain-Based Dynamic Key Management for Heterogeneous Intelligent Transportation Systems,” *IEEE Internet of Things Journal*, 4, 1832-1843, 2017.

[11] Shi J, Zeng X, Han R, “A Blockchain-Based Decentralized Public Key Infrastructure for Information-Centric Networks,” *Information*. 2022; 13(5):264. https://doi.org/10.3390/info13050264

[12] Patel M, “The security and privacy of wearable health and fitness Devices,” 2016. https://securityintelligence.com/the-security-and-privacy-of-wearable-health-and-fitness-devices. Accessed on October.

[13] Sabrina Sicari, Alessandra Rizzardi, Luigi Alfredo Grieco, and Alberto Coen-Porisini, “Security, privacy and trust in internet of things : The road ahead,” *Computer Networks*, 76 :146– 164, 2015.

[14] Yuchen Yang, Longfei Wu, Guisheng Yin, Lijie Li, and Hongbin Zhao, “A survey on security and privacy issues in internet-of-things,” *IEEE Internet of Things Journal*, 4(5) :1250–1258, 2017.

[15] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao, “A survey on internet of things : Architecture, enabling technologies, security and privacy, and applications,” *IEEE Internet of Things Journal*, 4(5) :1125–1142, 2017.

[16] https://www.certeurope.fr/blog/quest-ce-quune-pki-ou-infrastructure-a-cles-publiques/

[17] Koblitz, N, “Elliptic curve cryptosystems,” *Mathematics of Computation*, 48(177), 203-209, 1987.

[18] Victor S Miller, “Use of elliptic curves in cryptography,” In *Lecture Notes in Computer Sciences, on Advances in cryptology—CRYPTO 85, Springer-Verlag, New York, NY, USA*, 1986, 218, 417–426.

[19] Rafik, M. B. O. & Mohammed, F. (2013), "The impact of ECC’s scalar multiplication on wireless sensor network, " *2013 11th International Symposium on Programming and Systems (ISPS), Algiers, Algeria*, 2013, 17-23. doi:10.1109/ISPS.2013.6581488

[20] https://www.educative.io/answers/what-is-the-elliptic-curve-diffie-hellman-algorithm