Identity-Based Proxy Signature Scheme for IIoT Environment

Presented By
Pradhyumna Palore
2023202022
Ayush Rai
2023202020
Shiv Kumar Modi
2023202024

International Institute of Information Technology, Hyderabad

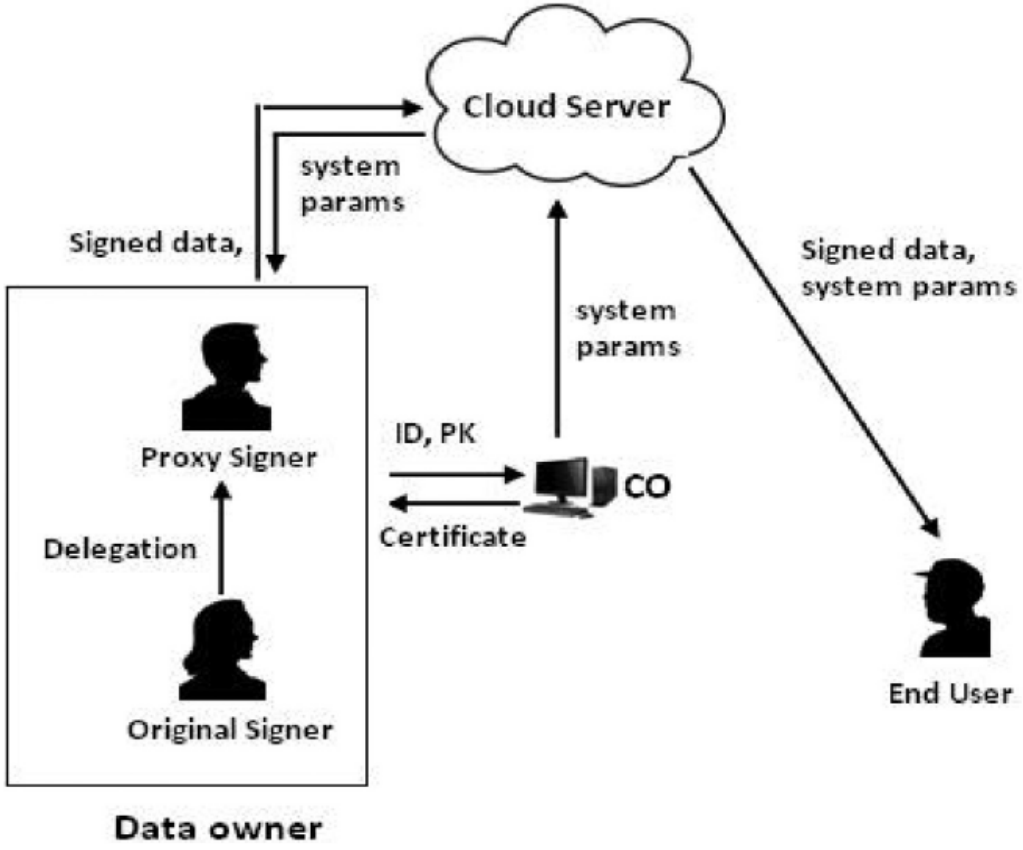
November 5, 2024

Outline

- Introduction
- Motivation
- Network Model
- Attack Model
- Proposed Solution
- Analysis
- Conclusion
- References

Introduction

- Focuses on secure signature schemes for IIoT device authentication.
- Analyzed certificate-based systems, highlighting key management challenges.



- Moving toward identity-based solutions for easier scalability in IIoT.

Motivation

- The Industrial Internet of Things (IIoT) requires real-time interactions among numerous devices in resource-constrained environments like manufacturing, autonomous vehicles, and smart infrastructure.
- certificate-based security is inefficient for IIoT due to high computational overhead, complex key management, and limited scalability.
- These limitations hinder IIoT devices, which need fast, secure communication while conserving processing power and energy.
- This proposal introduces a lightweight, scalable, and secure identity-based proxy signature scheme, eliminating the need for traditional certificates.
- By enhancing computational efficiency and simplifying key management, the scheme strengthens security, making it ideal for dynamic IIoT networks with frequent device interactions.

Attack Model

- **Man-in-the-Middle (MitM) Attacks:**
  - Attackers intercept and potentially alter data between IIoT devices and the central server.
  - Compromises data integrity and authenticity.
- **Key Exposure and Unauthorized Delegation:**
  - Exposure of proxy keys allows attackers to impersonate legitimate devices.
  - Enables unauthorized actions within the network, affecting device trust.
- **Replay Attacks:**
  - Adversaries capture and resend legitimate messages to disrupt communication.
  - Highlights the need for signature timestamping and revocation mechanisms.

Network Model

- **Cloud Server:**
  - Acts as a central hub for managing data processing and storage for IIoT devices.
- **Certification Authority (CA):**
  - A trusted entity responsible for generating certificates and keys for IIoT devices.
  - Ensures the authenticity of devices within the network.
- **Data Owner:**
  - The primary entity that authorizes other devices to act on its behalf.
  - Enables the delegation of signing capabilities to designated devices.
- **Proxy Signer (IIoT Device):**
  - Authorized, resource-constrained devices that perform proxy signing operations.
  - Verifies and forwards data securely on behalf of the Data Owner.

Proposal

- **Identity-Based Proxy Signature Scheme:**
  - Public keys are derived directly from device identities, removing the need for traditional certificates.
- **Efficient Multi-Signature Verification:**
  - Bilinear pairing enables rapid and scalable authentication, suited to resource-constrained IIoT environments.
- **Enhanced Security with CDH Assumption:**
  - Provides strong resistance against forgery and unauthorized access, ideal for dynamic IIoT networks.

Analysis of Proposal: Preliminaries

A. Bilinear Pairing

Let G_1 and G_2 be two groups of prime order q. A bilinear pairing is a map $e : G_1 \times G_1 \to G_2$ with the following properties:

1. **Bilinear:** $e(aP, bQ) = e(P, Q)^{ab}$, where $P, Q \in G_1$ and $a, b \in \mathbb{Z}_q^*$.
2. **Nondegenerate:** There exist $P, Q \in G_1$ such that $e(P, Q) \neq 1$.
3. **Efficiently Computable:** There is an efficient algorithm to compute $e(P, Q)$ for $P, Q \in G_1$.

B. Computational Diffie-Hellman Assumption

Given a group G_1 of prime order q with generator P and elements $aP, bP \in G_1$, where $a, b \in \mathbb{Z}_q^*$, the CDH problem in G_1 is to compute abP. We assume that the CDH problem over G_1 is hard.

Identity-Based Key Generation

In an identity-based key generation scheme, the Key Generation Center (KGC) computes the public and private keys for a user based on their identity ID_i. The key generation equations are as follows:

- **Public Key:**

$$Q_{\mathrm{ID}_i} = H_1(\mathrm{ID}_i) \quad \text{(where } H_1 \text{ is a hash function)}$$

- **Private Key:**

$$S_{\mathrm{ID}_i} = s \cdot Q_{\mathrm{ID}_i} \quad \text{(where } s \text{ is the master key of the KGC)}$$

The KGC securely sends the private keys to users.

- The original signer U_0 has: - Public Key: Q_{ID_0} - Private Key: S_{ID_0}
- Each proxy signer U_i (for $1 \leq i \leq n$) has: - Public and Private Keys: $(Q_{\mathrm{ID}_i}, S_{\mathrm{ID}_i})$.

Delegation Generation

To delegate signing capability to proxy signers, the original signer U_0 performs the following steps:

1. Generate a warrant message w that includes:
   - Identity information for the original signer and proxy group
   - Delegation period
   - Scope of proxy authority
   - Message types to be delegated
2. Randomly choose $r_0 \in \mathbb{Z}_q^*$ and compute:

$$R_0 = r_0 P$$

3. Compute:

$$h_0 = H_2(w, R_0) \quad \text{and} \quad V_0 = r_0 h_0 + S_{\mathrm{ID}_0}$$

4. Send $\sigma_0 = (R_0, V_0)$ and w to each proxy signer U_i for $1 \leq i \leq n$.

Delegation Verification

Each proxy signer U_i verifies the received delegation by performing the following:

- Compute:

$$h_0 = H_2(w, R_0)$$

- Check the verification condition:

$$e(P, V_0) = e(P_{\text{pub}}, Q_{\text{ID}_0}) \cdot e(R_0, h_0)$$

Proof Requirement: Show that if the equality holds, then U_i accepts σ_0 as a valid delegation.

$$e(P, V_0) = e(P, r_0 h_0 + S_{ID_0}) = e(P, r_0 h_0)\, e(P, S_{ID_0})$$
$$= e(P, r_0 h_0)\, e(P, s Q_{ID_0})$$
$$= e(P, h_0)^{r_0}\, e(P, Q_{ID_0})^{s}$$
$$= e(r_0 P, h_0)\, e(sP, Q_{ID_0}) \quad ; \quad \boxed{r_0 P = R_0} \; \boxed{sP = P_{pub}}$$
$$= e(R_0, h_0)\, e(P_{pub}, Q_{ID_0})$$

Proxy Key Generation

Each proxy signer U_i generates a proxy signing key using delegation (R_0, V_0) and their private key S_{ID_i}.

1. U_i selects a random secret $r_i \in \mathbb{Z}_q^*$, computes:

$$R_i = r_i P$$

   and broadcasts R_i as a public proxy key.

2. Compute the proxy signing key:

$$S_{U_i} = r_i h_0 + S_{\mathrm{ID}_i} + V_0$$

   where $h_0 = H_2(w, R_0)$.

3. Publish the public parameter list:

$$(R_i, R_0, w)$$

   for proxy signing key S_{U_i}.

Multiproxy Signing

Given a message m to be signed, each proxy signer U_i (for $1 \leq i \leq n$) performs the following steps:

1. Chooses a random secret $x_i \in Z_q^*$, computes the partial public key:

$$X_i = x_i P$$

   and broadcasts X_i to the proxy group.
2. Each proxy signer computes the aggregate public key $X = \sum_{i=1}^{n} X_i$.
3. Computes a partial proxy signature V_i for $h = H_3(m, X)$:

$$V_i = x_i h + S_{U_i}$$

4. Sends the partial proxy signature $\sigma_i = (w, R_0, R_i, X_i, V_i)$ to the Master.

Multiproxy Signature Verification

Upon receiving each partial proxy signature $\sigma_i = (w, R_0, R_i, X_i, V_i)$ from each proxy signer U_i, the Master verifies the validity of each partial proxy signature as follows:

1. Computes $h = H_3(m, X)$ and checks the verification condition:

$$e(P, V_i) = e(P_{\text{pub}}, Q_{\text{ID}_0} + Q_{\text{ID}_i}) \cdot e(X_i, h) \cdot e(R_0 + R_i, h_0)$$

Proof Requirement: Show that if the equality holds, then σ_i is a valid partial multiproxy signature.

$$e(P, V_i) = e(P, x_i h + S_{V_i}) = e(P, x_i h + r_i h_0 + s_{ID_i} + V_0)$$
$$= e(P, x_i h) \cdot e(P, r_i h_0) \cdot e(P, s Q_{ID_i}) \cdot e(P, r_0 h_0 + s Q_{ID_0})$$
$$= e(x_i P, h)\, e(r_i P, h_0)\, e(sP, Q_{ID_i})\, e(r_0 P, h_0)\, e(sP, Q_{ID_0})$$
$$= e(X_i, h)\, e(R_i, h_0)\, e(P_{pub}, Q_{ID_i})\, e(R_0, h_0)\, e(P_{pub}, Q_{ID_0})$$
$$= e(X_i, h)\, e(R_i + R_0, h_0)\, e(P_{pub}, Q_{ID_i} + Q_{ID_0})$$

Multiproxy Signature Verification (cont.)

If the verification condition holds for each proxy signer U_i, the Master aggregates the results to produce the final multiproxy signature:

2. Aggregates the components:

$$V = \sum_{i=1}^{n} V_i \quad \text{and} \quad R = \sum_{i=1}^{n} R_i$$

3. Generates the final multiproxy signature for the message m as:

$$\sigma = (w, R_0, R, X, V)$$

Multiproxy Signature Verification

The Multiproxy Verification (Multiproxy-Ver) allows anyone to verify the received multiproxy signature $\sigma = (w, R_0, R, X, V)$ for a message m as follows:

1. Given $\{\mathrm{ID}_j\}_{j \in M}$, m, and σ, the verifier checks if the message m conforms to the warrant w.
   - If m does not conform to w (e.g., message types not delegated in w), the verification fails.
2. The verifier checks if the n proxy signers $U_1, U_2, \ldots, U_n$ are correctly listed in the warrant w.
   - If they are not members in w, the verification fails.
3. Finally, the verifier accepts σ if and only if the following condition is satisfied:

$$e(P, V) = e\left(P_{\mathrm{pub}}, nQ_{\mathrm{ID}_0} + \sum_{i=1}^{n} Q_{\mathrm{ID}_i}\right) \cdot e(X, h) \cdot e(R + nR_0, h_0)$$

Multiproxy Signature Verification - Proof

Verification Proof: To ensure that the received multiproxy signature $\sigma = (w, R_0, R, X, V)$ is valid, the following proof illustrates the verification process.

$$e(P, V) = e\left(P, \sum_{i=1}^{n} (x_i h + S_{U_i})\right)$$

$$= e\left(P, \sum_{i=1}^{n} x_i h\right) e\left(P, \sum_{i=1}^{n} S_{U_i}\right)$$

$$= e\left(\sum_{i=1}^{n} x_i P, h\right) e\left(P, \sum_{i=1}^{n} (r_i h_0 + S_{ID_i})\right) e(P, \Sigma V_0)$$

$$= e(X, h)\, e(P, \Sigma r_i h_0)\, e(P, s\, \Sigma Q_{ID_i})\, e(P, n V_0)$$

$$= e(X, h)\, e(\Sigma r_i P, h_0)\, e(sP, \Sigma Q_{ID_i})\, e(P, n(r_0 h_0 + s Q_{ID_0}))$$

$$= e(X, h)\, e(R_i, h_0)\, e(P_{pub}, \Sigma Q_{ID_i})\, e(nR_0, h_0)\, e(P_{pub}, nQ_{ID_0})$$

$$= e(X, h)\, e(R_i, h_0)\, e(P_{pub}, \Sigma Q_{ID_i})\, e(nR_0, h_0)\, e(P_{pub}, nQ_{ID_0})$$

$$= e(X, h)\, e(nR_0 + R_i, h_0) \cdot e\left(P_{pub}, nQ_{ID_0} + \sum_{i=1}^{n} Q_{ID_i}\right)$$

Conclusion

- **Enhanced Efficiency:**
  - Identity-based bilinear pairing eliminates the need for certificates, reducing computational load and management complexity.
  - Achieves $O(1)$ complexity for key generation and verification, optimizing performance for resource-constrained IIoT devices.
- **Scalability:**
  - Supports aggregation of multiple signatures, enabling single-step verification.
  - Ideal for large-scale IIoT networks with frequent device interactions and dynamic changes.
- **Strong Security:**
  - Based on the Computational Diffie-Hellman (CDH) assumption, ensuring resistance to forgery and unauthorized access.
  - Isolates proxy signing keys from original keys, enhancing protection in delegated tasks.

References

- Girraj Kumar Verma, B.B. Singh, Neeraj Kumar, Mohammad S. Obaidat, Debiao He, Harendra Singh, "An efficient and provable certificate-based proxy signature scheme for IIoT environment," *Information Sciences*, 2020.
- Ahmad-Reza Sadeghi, Christian Wachsmann, Michael Waidner, "Security and privacy challenges in industrial internet of things," *Proceedings of the 52nd Annual Design Automation Conference*, 2015.
- Dennis Meffert, "Bilinear Pairings in Cryptography," MSc Thesis, Radboud University Nijmegen, 2007.
- Girraj Kumar Verma, B.B. Singh, "Short certificate-based proxy signature scheme from pairings," *Transactions on Emerging Telecommunications Technologies*, 2020.
- Girraj Kumar Verma, B.B. Singh, Harendra Singh, "Provably secure certificate-based proxy blind signature scheme from pairings," *Information Sciences*, 2018.

Thank You!

Contact us at: *pradhyumna.palore@students.iiit.ac.in*