**White Paper: Streamlining Device Authentication: A White Paper on MTLS Implementation**

**Version 1.0**

**Date: September 12, 2025**

**Contributors**


- Avani Patel
- Cabre Eduardo
- Masud Chowdhury
- Pravin Chaudhari
- Piedrahita Sergio
- Adam Mitchell
- Tom Dodson


## 1. **Executive Summary**


This white paper outlines the architecture and implementation of MTLS-based client authentication for edge and IoT devices, eliminating the need for traditional username and password management. The solution leverages client TLS certificate issuance for device identity and rotation, enhancing security and simplifying device provisioning processes.


## 2. Glossary of Terms

| Term | Definition |
|---|---|
| CSS API | The API that processes CSRs and issues certificates |
| Bootstrapping Issuing CA | The Certificate Authority that issues Bootstrapping Certificates |
| Bootstrapping Certificate | Certificate used by the client to authenticate and obtain a Client TLS Certificate |
| Client TLS Certificate | The client certificate is used by the device to establish a Mutual TLS session with certain MTLS enabled PSS APIs. |
| MTLS | Mutal TLS. An optional client authentication variation of the TLS protocol. |
| C2PA | Coalition for Content Provenance and Authenticity. An industry standard defining protocol for proof of digital authenticity. |
| Application PKI | The CA chain deployed in the system used by the Customer to issue certificates for a product security feature (i.e. DICE, C2PA, TPM, etc.) |

| | |
|---|---|
| PSS SDK | A client library that facilitates interacting with PSS service APIs. |

## 3. Introduction

As technology continues to advance, secure device authentication becomes increasingly crucial, particularly for the growing number of edge and IoT devices. This white paper introduces a robust MTLS-based authentication system designed to replace traditional password management by using client TLS certificates for device identity and rotation. This approach simplifies device provisioning and strengthens security, addressing significant challenges faced by manufacturers and service providers.

Additionally, this paper examines alternative methods to enhance security against potential threats like BORE attacks, offering a comprehensive strategy for device protection. The paper aims to improve security protocols, streamline authentication processes, and provide practical insights for implementation.

## 4. High-Level Design Objectives

- **Eliminate Passwords**: Remove the need for password management during manufacturing and deployment.
- **Facilitate Device Manufacturing**: Enable easy programming of Bootstrapping Certificates into devices during manufacturing.

  **Initial Device Setup: Use Bootstrapping Certificates for obtaining device-specific Client TLS Certificates**

## 5. Solution Architecture

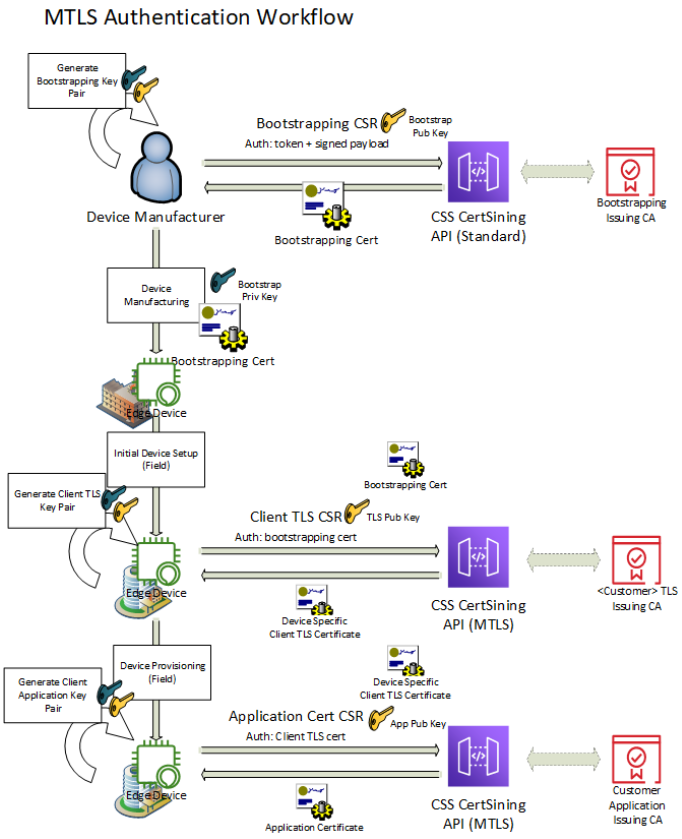


*Figure 1 - MTLS Authentication Workflow*

### 1.1. Authentication Workflow

Customers are created and granted access to the system, which supports multiple authentication methods.

a. The system creates bootstrapping and mTLS CAs for the Application PKI Hierarchy.
b. The system grants customers access to bootstrapping and mTLS CAs through a unique user policy, ensuring that only the designated user has access to the Application PKI Hierarchy. This hierarchy includes the CA chain used to issue device certificates for security features such as DICE, C2PA, and TPM EK.
c. Customer generates the Bootstrapping Key Pair
d. Customer constructs CSR with the Bootstrapping Public Key and submits to the standard Signing Service API
    i. Customer uses Cognito OAuth token and Payload Signature to authenticate

e. System adds Subject Altname extension containing the Customer's User ID to the bootstrapping certificate and returns to customer
f. Customer receives the Bootstrapping Certificate from CSS
g. Customer provisions the Bootstrapping Private Key and Certificate in device
h. Device is deployed and Initial Device Setup is initiated
i. Device generates the Client TLS Key Pair
j. Device constructs the Client TLS CSR with Client TLS Public Key and submits to the MTLS protected Signing Service API using the Bootstrapping Certificate to authenticate to API
    i. Device uses Bootstrapping Certificate to authenticate to API
k. System adds Subject Altname extension containing the Customer's User ID to the Client TLS Certificate and returns to device
l. Customer receives Client TLS Certificate from CSS
m. Device Provisioning of the device starts
n. Device generates the Application Key Pair
o. Device constructs the Application CSR with the Application Public Key and submits to the MTLS protected Signing Service API using the Client TLS Certificate to authenticate to the API
    i. Device uses Client TLS Certificate to authenticate to the API
p. Client receives the Application Certificate

### 1.2. Alternative Workflow

The Customer may implement an alternative workflow if they want to limit exposure of the Bootstrapping Keypair. Exposure of the Bootstrapping Keypair could potentially lead to a BORE attack. To mitigate this risk, the Customer may decide to provision the device-specific Client TLS Certificate during manufacturing using a special Manufacturing Tool responsible for obtaining the individual Client TLS Certificates as follows:

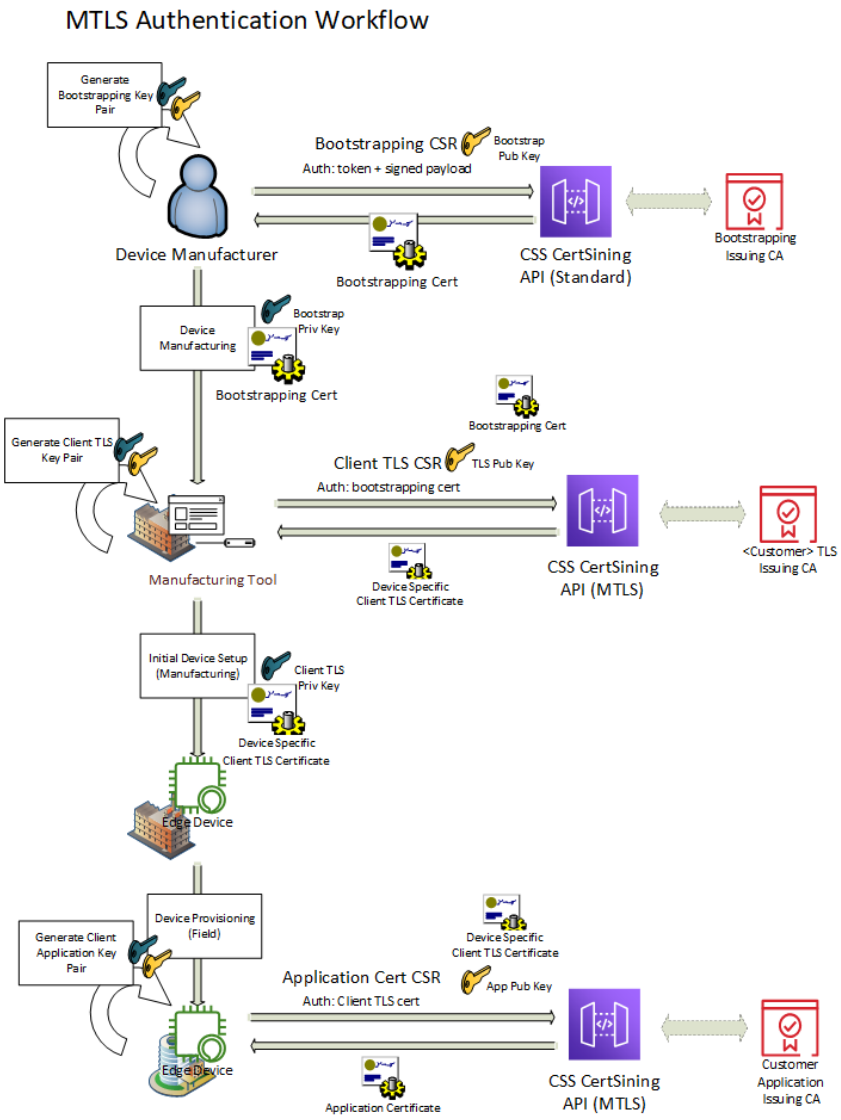


*Figure 2 Alternative MTLS Workflow*

### 1.3. Bootstrapping Certificate

The Bootstrapping Certificate is a global certificate that the Customer provisions into their devices during manufacturing.

#### 3.2.1 Bootstrapping certificate chain design

3.2.1.1 **TLS Root CA:** A Root CA is created in the PSS System. This Root CA will be shared among all customers and will serve as the trust anchor for the Bootstrapping and the Client TLS Cert Chains.

3.2.1.2 **Bootstrapping Intermediate CA:** A single Intermediate CA anchored in the Root shall be created and deployed into CSS in the PSS system. This Intermediate CA shall be rotated annually and shall have a 2-year validity period.

3.2.1.3 **Bootstrapping Issuing CA:** This CA will be anchored to the Intermediate CA and will be responsible for issuing the customer's Bootstrapping Certificates. This is a single CA shared across all customers. This CA's certificate shall be valid for 2 years, and the key shall be rotated every 90 days.

3.2.1.4 **Bootstrapping Leaf Certificate:** The device manufacturer shall generate a Bootstrapping key pair and submit a CSR to the Issuing CA in order to get the Bootstrapping Certificate for these devices. This certificate will have a validity period of 1 year.

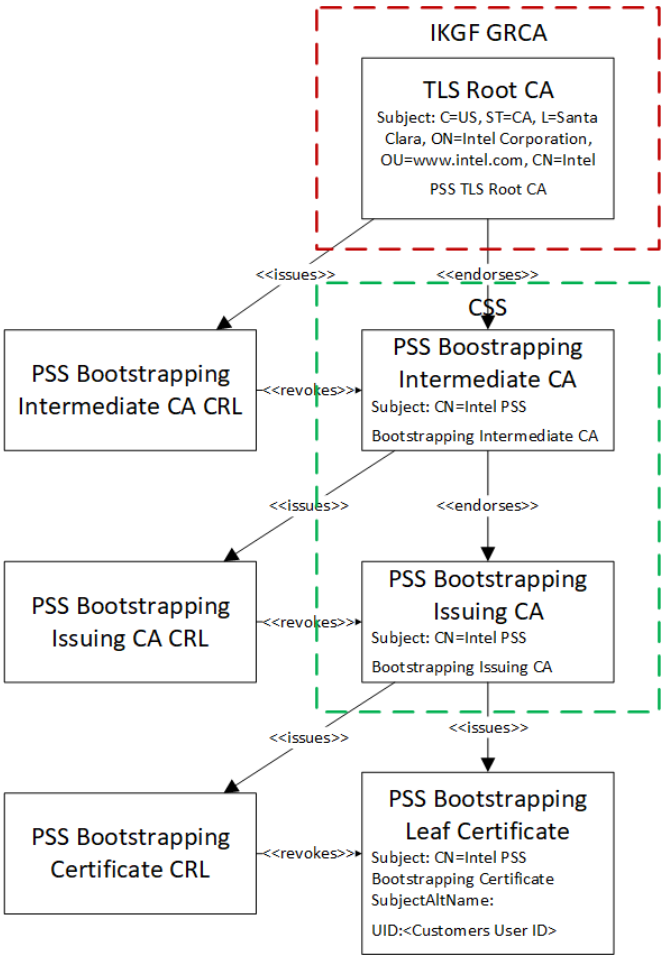


*Figure 3: PSS Bootstrapping Certificate Hierarchy*

### 3.2.2 Bootstrapping Issuing CA Access Policy

The Client must be an established customer of the System. The Client must present access token and valid JSON payload including PSS Bootstrapping Leaf certificate CSR and signature.

### 3.2.3 Obtaining the Bootstrapping certificate

Upon completion of customer onboarding the customer will receive its Cognito credentials. The customer's payload signing public key would have been registered in the system. The customer will proceed to create a bootstrapping key pair and send the Bootstrapping CSR containing its public key to the PSS Signing Service API. This request will be submitted to the Bootstrapping Issuing CA CAID. CSS issues the customer's Bootstrapping Certificate and sends back as the response.

### 3.2.4 Bootstrapping certificate rotation

Generation of a new Bootstrapping Key is required for each new Bootstrapping Certificate request. If the Bootstrapping Certificate expires, the customer shall obtain a new certificate from PSS. Client TLS Certificate. To facilitate edge and IOT device access to PSS services, certain service API endpoints will support MTLS Client Authentication. These endpoints will not require the client to present a Cognito token or payload signing. To obtain the Client TLS Certificate, the device shall implement a provisioning mechanism such that the Client TLS Certificate is obtained prior to initializing the device. This process is identified in Figure 1 as “Initial Device Setup”. It is the responsibility of the device manufacturer to implement the firmware/software required to obtain the Client TLS Certificate. The CSS SDK will facilitate this process.

3.2.3.1 **Client TLS Certificate Chain Design** TLS Intermediate CA: A single Intermediate CA anchored in the Root shall be created and deployed into CSS.

This Intermediate CA shall be rotated annually and shall have a ten-year validity period.

3.2.3.2 **Customer TLS Issuing CA:** An issuing CA shall be deployed for each Customer intermediate CA and will be responsible for issuing the customer's Client TLS Certificates.

3.2.3.3 **Client TLS Leaf Certificate**: The device shall generate a Client TLS Key Pair and submit to the CSS Cert-Signing API to obtain its leaf certificate. This certificate will have a validity period of 1 year,

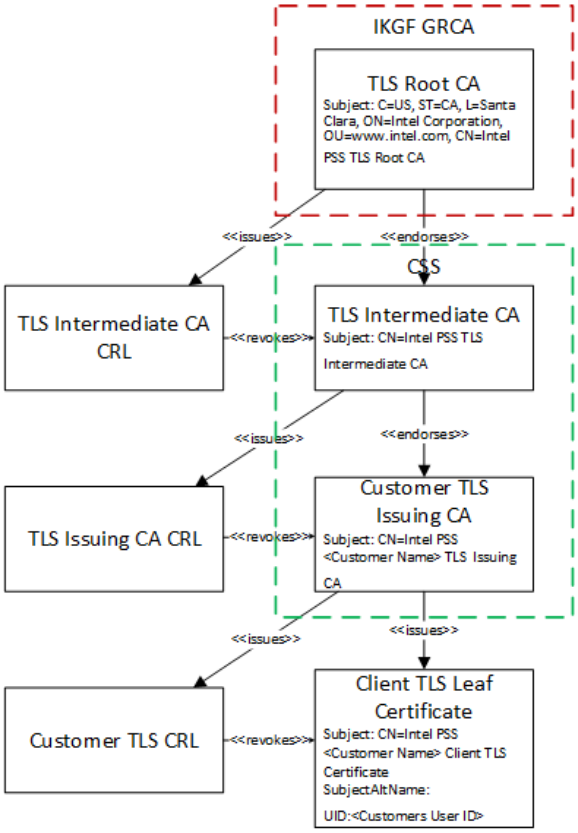


*Figure 4 - Client TLS PKI Hierarchy*

### 3.2.4 Customer TLS Issuing CA Access Policy

The Client must present a valid Client TLS Leaf Certificate issued by IKGF for mTLS. The Client certificate must be signed by Customer TLS Issuing CA. The Client TLS Leaf Certificate must include Customer user ID in Subject Alt Name and Subject name should have Intel PSS <Customer Name> Client TLS Certificate. The Client certificate must be valid and not expired. Client Certificate must be verified against the trusted

TLS certificate chain in the system. The Client should be configured to use TLS (version that are supported).

#### 3.2.5 Client certificate rotation

The Client TLS Key Pair shall be rotated every 30 days, and a new Client TLS Certificate must be obtained. If rotation is not possible within 30 days, the device shall rotate the key and obtain a new certificate during device initialization.

#### 3.2.6 Client certificate revocation

The Client TLS Certificate may be revoked by the customer using the CSS revocation capability. Since the customer owns their TLS Issuing CA, they will have the ability to individually revoke Client TLS Certificates. If the customer suspects that all devices are compromised, they may request IKGF to revoke the customer's TLS Issuing CA certificate, thereby invalidating all their devices. The customer will need to repeat the Initial Device Setup to provision a new Bootstrapping Certificate for each device.

### 3.3 MTLS Authentication

#### 3.3.1 Integration with AWS API Gateway

To enable mutual TLS in API Gateway a new regional custom domain shall be created. This domain is configured with mutual authentication and associated with the Trust store file. Since domains exist that are not mutual TLS protected, we shall create new regional domains exclusively for endpoints with MTLS authentication.

#### 3.3.2 API Gateway Trust Anchor Updates

The API Gateway forwards the Client TLS Certificate to the Lambda authorizer, which parses the certificate to extract the Subject Altname extension and retrieve the User ID from the UID field.

#### 3.3.3 Lambda authorizer for identity/params extraction

The API Gateway forwards the Client TLS Certificate to the Lambda authorizer, which parses the certificate to extract the Subject Altname extension and retrieve the User ID from the UID field.

### 3.4 Authorization Policy Application

Systems rely on internal access management by applying the necessary access policies to client requests. This solution should not require modification of existing access control policy mechanisms.

## 4 PSS Client SDK

### 4.1 Client TLS Certificate and Key Storage

The PSS SDK will support enabling MTLS authentication and connectivity to the newly created MTLS enabled endpoints.

## 5 Use Cases:

### 6.1 Enhancing Edge Device Authentication for Advanced PSS Capabilities:

In the rapidly evolving landscape of edge and IoT devices, direct authentication is crucial for delivering advanced capabilities. MTLS-based authentication, as facilitated by PSS services, offers a secure and efficient method for managing device interactions. A key application is within the Coalition for Content Provenance and Authenticity (C2PA), where frequent issuance of signing certificates for media capture devices is necessary to maintain content integrity.

PSS services can efficiently implement MTLS-based authentication to handle these requirements, ensuring secure communication and integration across devices. This approach not only simplifies the authentication process but also supports scalability, allowing systems to adapt to future technological advancements and increased device proliferation. By leveraging PSS services, organizations can enhance their security posture and streamline operations in the edge and IoT device ecosystem.

### 6.2 Enhancing Industrial IoT Security:

In industrial environments, IoT devices like sensors and actuators play a critical role in monitoring and controlling machinery and processes. MTLS-based authentication offers a secure communication method for these devices, reducing the risk of cyberattacks that could disrupt operations or compromise safety. By adopting this authentication model, industries can maintain operational integrity and ensure the safety of their systems, even as they integrate more IoT devices into their workflows.

### 6.3 Improving Supply Chain Management:

IoT devices are increasingly used to track and manage inventory and shipments in real-time across global supply chains. MTLS-based authentication ensures that data from these devices is accurate and secure, improving transparency and efficiency in supply chain operations. This approach not only enhances the reliability of supply chain data but also supports better decision-making and resource management, helping businesses optimize their logistics processes.

## Conclusion

The MTLS-based client authentication architecture provides a robust solution for edge and IoT device security, simplifying provisioning processes and enhancing overall security posture. By leveraging AWS MTLS support and eliminating password management, this approach aligns with modern security requirements and facilitates seamless device integration into PSS services.

## Appendices

- **Appendix A**: Detailed Certificate Chain Designs
- **Appendix B**: API Gateway Configuration Steps
- **Appendix C**: SDK Implementation Guidelines

## References


- AWS API Gateway Documentation
- NIST Cryptographic Standards
- Coalition for Content Provenance and Authenticity (C2PA) Standards