**Tel**: +27 11 343 2000 | **Fax**: +27 11 442 5908 | **Email**: info@prism.co.za
**Address**: President Place, Johannesburg, South Africa

# PrismToken User Guide

2022/02/15

| | |
|---|---|
| **Document number:** | PR-D2-1095 Rev 1.5 |
| **Release date:** | 2022/02/15 |
| **Prepared by:** | SG, TD,CA |
| **Copyright:** | © 2022 Prism Payment Technologies (Pty) Ltd. |
| **Synopsis:** | Guide for key management and operation of PrismToken STS6. (TSM250 & TSM500i-NSS) |

## Company Confidential

*The information in this document is intended only for the person or the entity to which it is addressed and may contain confidential and/or privileged material. Any views, recreation, dissemination or other use of or taking of any action in reliance upon this information by persons or entities other than the intended recipient, is prohibited.*

## Disclaimer

*Prism Payment Technologies (Pty) Ltd. makes no representations or warranties whether expressed or implied by or with respect to anything in this document, and shall not be liable for any implied warranties of merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.*

# Contents

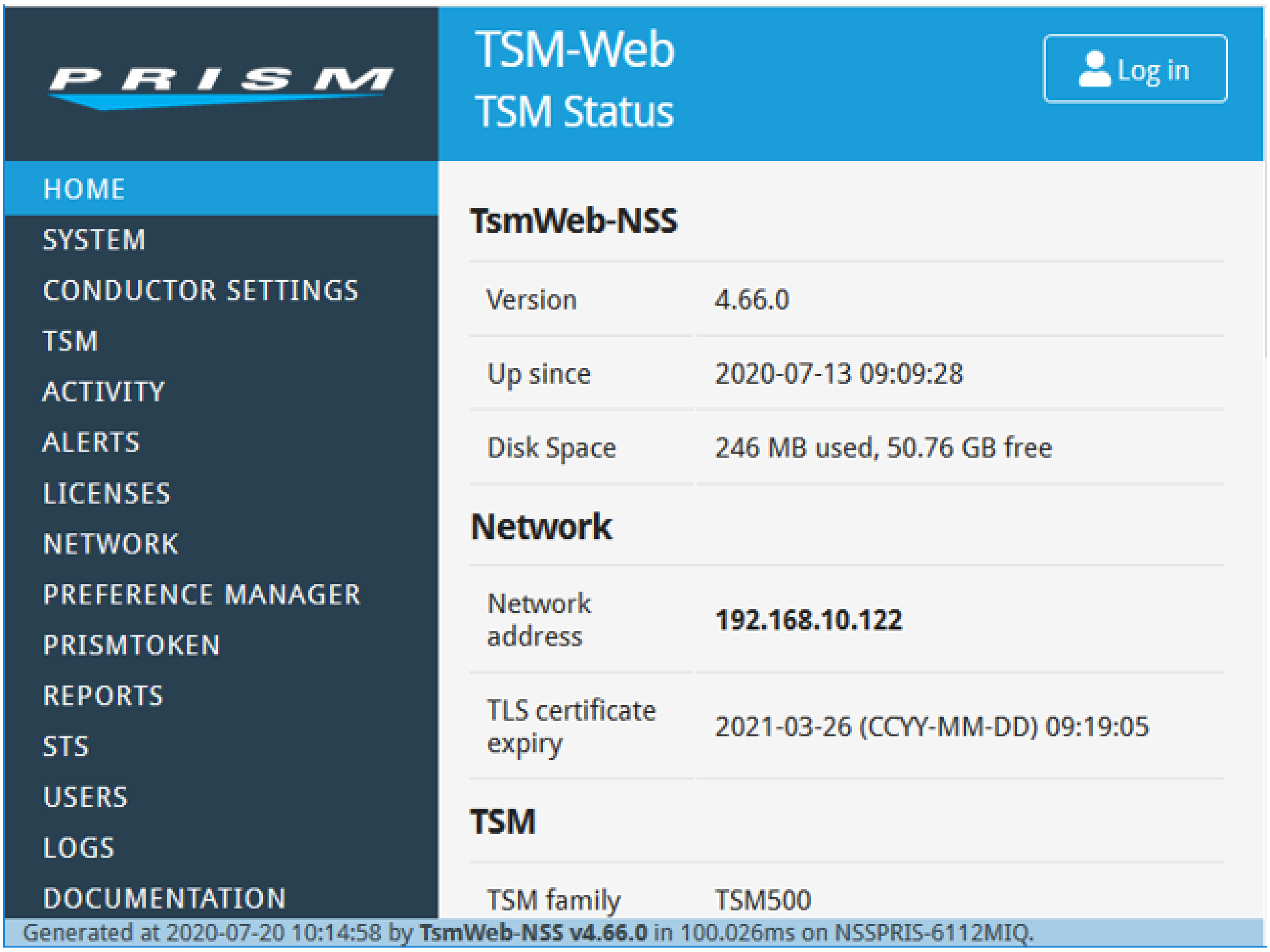
# Figures

# 1 What is PrismToken?

For an overview of what PrismToken is and what is does refer to Introduction to STS and PrismToken (PR-D2-1060) document.

# 2 Home

Using a web browser enter the link for the configured TSM-Web and press enter. The first tab that you will be directed to is the Home tab. On the home tab you will find useful information relating to the status of the NSS and services running on the NSS.



TsmWeb-NSS

| | |
|---|---|
| Version | 4.66.0 |
| Up since | 2020-07-13 09:09:28 |
| Disk Space | 246 MB used, 50.76 GB free |

Network

| | |
|---|---|
| Network address | **192.168.10.122** |
| TLS certificate expiry | 2021-03-26 (CCYY-MM-DD) 09:19:05 |

TSM

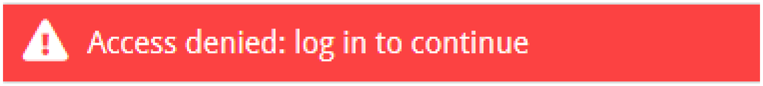
| | |
|---|---|
| TSM family | TSM500 |

Generated at 2020-07-20 10:14:58 by **TsmWeb-NSS v4.66.0** in 100.026ms on NSSPRIS-6112MIQ.

**Figure 1 Home tab**

# 3 Login

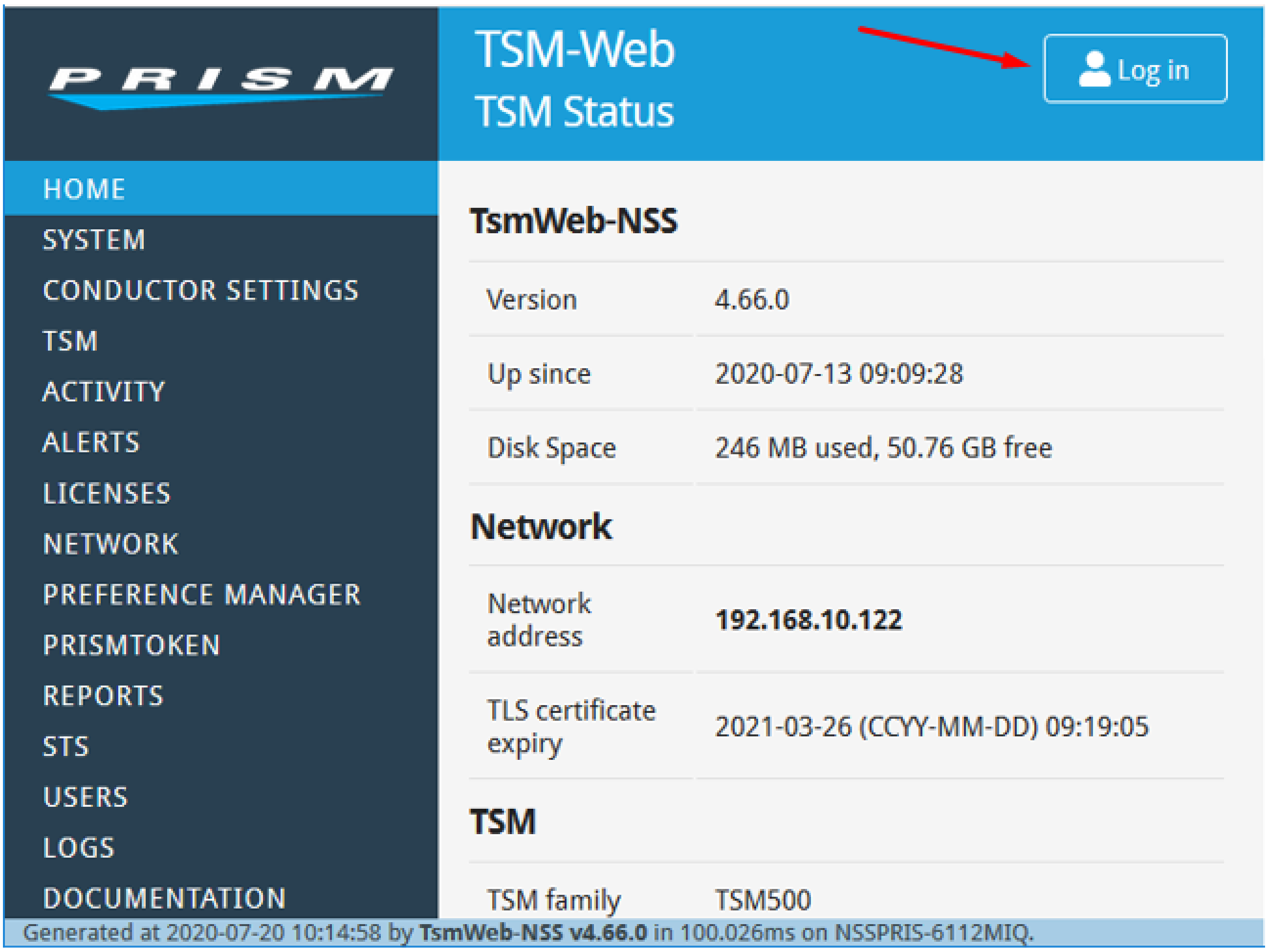
To gain access to the remaining tabs the user requires a username and password. With the exclusion of the Documentation tab the remaining tabs will display an error if the user doesn't login and clicks on them.



Access denied: log in to continue

Figure 2 Access denied

The user can access the login screen by clicking on the Login button located at the top right corner on TSM-Web. Users can be created on TSW-Web, please refer to the TSM500i and TSM-Web User Guide on how to add and manage users.



TSM-Web
TSM Status

TsmWeb-NSS

| | |
|---|---|
| Version | 4.66.0 |
| Up since | 2020-07-13 09:09:28 |
| Disk Space | 246 MB used, 50.76 GB free |

Network

| | |
|---|---|
| Network address | **192.168.10.122** |
| TLS certificate expiry | 2021-03-26 (CCYY-MM-DD) 09:19:05 |

TSM

| | |
|---|---|
| TSM family | TSM500 |

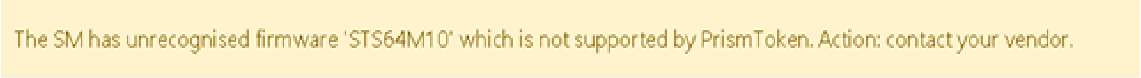
Generated at 2020-07-20 10:14:58 by **TsmWeb-NSS v4.66.0** in 100.026ms on NSSPRIS-6112MIQ.

Figure 3 Login button

# 4 Installation and Initial Configuration

### 4.1.1 For Manufacturing

- Install and configure your Prism TSM500i-NSS Security Module
    a) Refer to "TSM500i and TsmWeb User Guide" (PR-D2-0854).
    b) You will need password reset certificates to set up Crypto Officer passwords. Contact your vendor if you have not received these certificates.
- At this stage you should have access to the following:
    a) Access to the Security Module Web Interface (TsmWeb) via a web browser.
    b) Credentials to log in to the Web Interface as an admin of TsmWeb.
    c) Crypto Officer passwords that have been set up using Password Reset certificates obtained from Prism.
- After changing the IP address of your NSS you must regenerate the SSL/TLS certificate so that it has the correct details, reboot the TSM500i-NSS, and restart your web browser.
- Switch PrismToken to "Meter Manufacturing" mode:
    - If your SM has Manufacturing firmware but you do not switch TsmWeb to "Meter Manufacturing" mode, you may see an error like this:



The SM has unrecognised firmware 'STS64M10' which is not supported by PrismToken. Action: contact your vendor.

    - Log in to TsmWeb using the admin credentials.
    - Grant the "ptoken-admin" permission to the TsmWeb "admin" account. Refer to 5.2 Users for instructions.
    - Click on "Preference Manager" in the left navigation menu. Find preference "ptoken.meterManufacturingMode", click "Edit" and set the value to True.
    - After changing the preference you must restart TsmWeb, for an NSS you can do this via the "System" link in the left navigation menu.
- The DITK (also known as the ROM Dispenser Key) will need to be loaded as part of the initial configuration of PrismToken. Refer to section 6 DITK.

### 4.1.2 For Vending

- Install and configure your Prism Security Module and the Tsm-Web Interface:
    a) For a TSM500i-NSS refer to "TSM500i and TsmWeb User Guide" (PR-D2-0854). For a vending environment you do not require Crypto Officer passwords and you do not need to load an SMK or other CSPs.

b) For a TSM250 refer to “TSM250 Installation and User Guide” (PR-D2-0988), also install TsmWeb per “TSM500i and TsmWeb User Guide” (PR-D2-0854).

- After changing the IP address of your must regenerate the SSL/TLS certificate so that it has the correct details. For the TSM500i-NSS reboot the NSS and for the TSM250 restart the TsmWeb Windows service. Restart your web browser.
- At this stage you should have the following:

  a) Access to the Security Module Web Interface (TsmWeb) via a web browser.

  b) Credentials to log in to the Web Interface as an admin (of TsmWeb).

# 5 PrismToken

## 5.1 PrismToken License Certificate

While logged in to the Security Module Web Interface (TsmWeb) as an admin: If you don’t see a “PRISMTOKEN” left menu item, load the PrismToken license using the “License” tab of the Web Interface, then restart TsmWeb.

Whilst logged into TsmWeb as an administrator you should see the “PRISMTOKEN” tab in the left navigation bar. If the PrismToken tab isn’t visible then you will need to load the PrismToken license using the “LICENSE” tab in the left navigation bar. A PrismToken license can be acquired from your vendor. Once the PrismToken license has been loaded from a PC restart TsmWeb from the services control panel and you should now have the “PRISMTOKEN” tab available.

## 5.2 Users

- Login to TsmWeb as an administrator and click on the “USERS” tab.

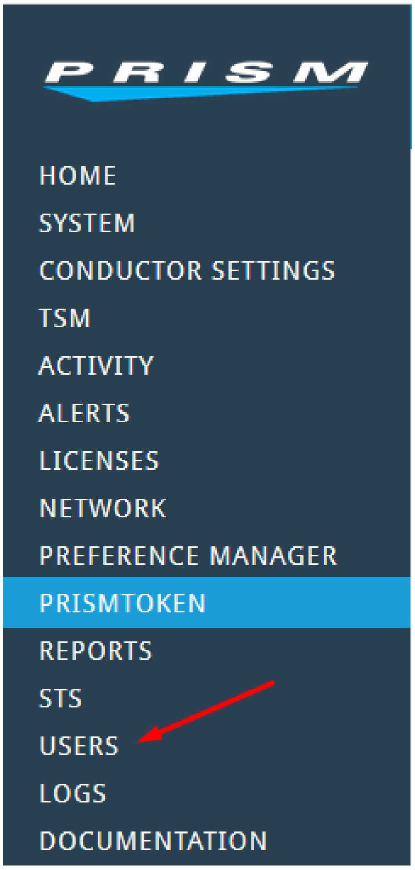
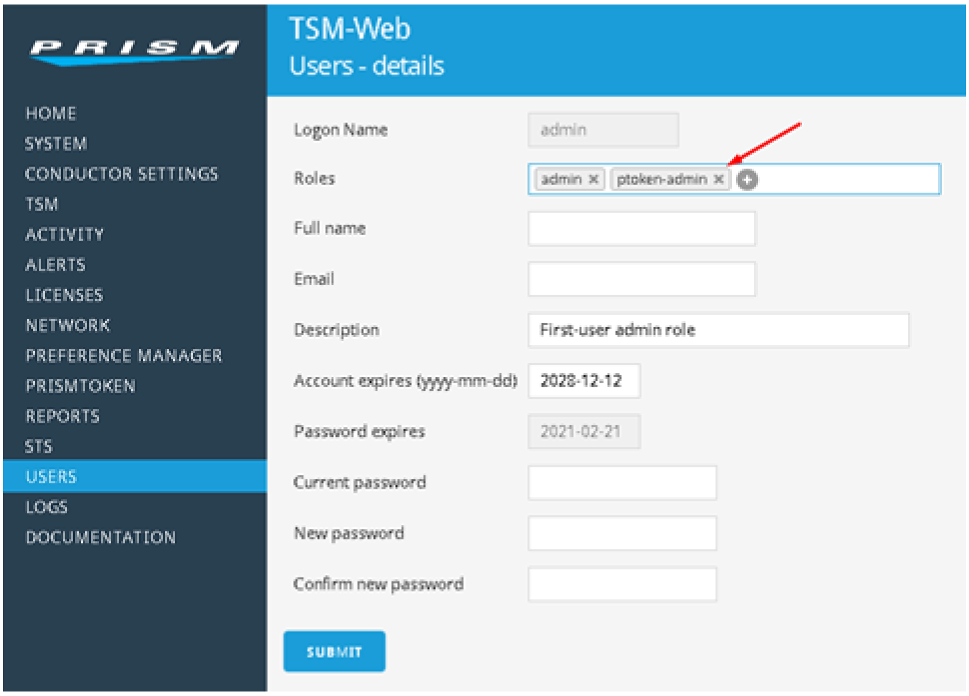


Figure 4 User Tab

- It’s possible to add roles to existing users or create new users and assign roles to those newly created users.
- You will require the following users

1. A new or existing user account with the 'ptoken-admin' role. Users with this role can administer and operator PrismToken. Administration includes setting preferences, and associating PrismToken with one or more KMCs.
2. For each individual who will operate PrismToken, should have their own user account with the 'ptoken-operator' role. Operation includes getting keys from the KMC and managing Vending Keys.
   a. For each user account the rights should be suitably limited.
   b. For technical operators, the user account permissions should be limited to generation of key change tokens and eningeering tokens.
3. Separate user account(s) for testing can be used during development and integration testing. These accounts should be disabled when your PrismToken system is deployed in a live environment.
4. For each client system that will use PrismToken via the Thrift API, a new user account with the 'ptoken-api' role plus one or more 'ptoken-issue-*' roles. For security reasons these accounts should not have any other TsmWeb roles (that would allow them to log in to TsmWeb via the Web Interface).



TSM-Web
Users - details

Logon Name: admin
Roles: admin × ptoken-admin ×
Full name:
Email:
Description: First-user admin role
Account expires (yyyy-mm-dd): 2028-12-12
Password expires: 2021-02-21
Current password:
New password:
Confirm new password:

Figure 5 ptoken admin role

## 5.3 Preferences

- PrismToken preferences can be found in the "PREFERENCE MANAGER" tab on the left navigation bar.
- PrismToken preferences contain a "ptoken" prefix.
- The default preferences should be suitable for vending.

- To issue proprietary engineering tokens (class=2, subclass=11-15) you must set preference 'ptoken.allowIssueProprietaryTokens' to 1 the default is 0, and the user account must have role 'ptoken-issue-any' or both 'ptoken-issue-eng' + 'ptoken-issue-proprietary'.

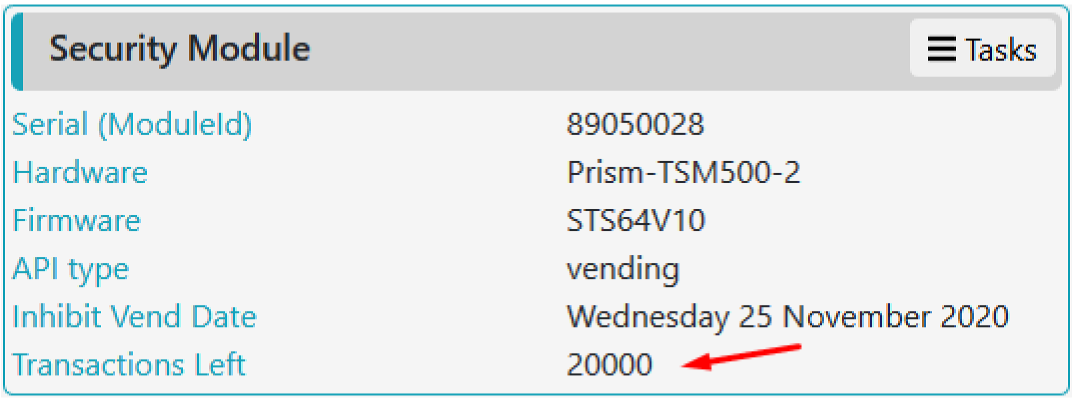
## 5.4 Dashboard

To access the PrismToken dashboard you will need to be logged in as a PrismToken administrator and click on the "PRISMTOKEN" tab in the left navigation bar.

On the dashboard you will find a "Notices" window. You will need to attend to all of the notices that you may have.

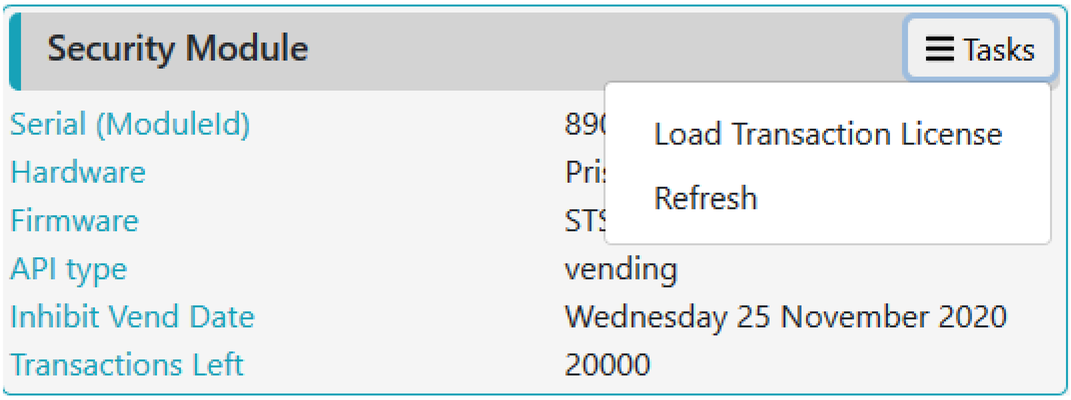
### 5.4.1 Transactions left

If "Transactions Left" is 0 or the "Inhibit Vend Date" is in the past' then you will need to load a Transaction License. Contact your vendor to obtain the Transaction License if required. To load the license on the PrismToken Dashboard, in the "Security Module" box, click the "Tasks" menu and select "Load Transaction License". Follow the on-screen instructions. Please note that the TX license is applicable to STS64Vxx firmware only(STS6 vending HSMs).



| | |
|---|---|
| Serial (ModuleId) | 89050028 |
| Hardware | Prism-TSM500-2 |
| Firmware | STS64V10 |
| API type | vending |
| Inhibit Vend Date | Wednesday 25 November 2020 |
| Transactions Left | 20000 |

Figure 6 Transactions left



| | |
|---|---|
| Serial (ModuleId) | 89( |
| Hardware | Pri |
| Firmware | STS |
| API type | vending |
| Inhibit Vend Date | Wednesday 25 November 2020 |
| Transactions Left | 20000 |

Figure 7 Load Transaction License

## 5.5 KMC PUBKEYs

You may need to load one or more KMC PUBKEYs. These keys are required before PrismToken can obtain Vending Keys from the KMC. You must request the KMC PUBKEY from your STS Key Management Centre. For testing purposes you can find the KMC PUBKEY for Prism's test KMC under the "Documentation" link in TsmWeb's menu.

- An admin user account is required to load a public key. Using your admin user login to TsmWeb.
- Click on "Preference Manager" tab in the left navigation bar.

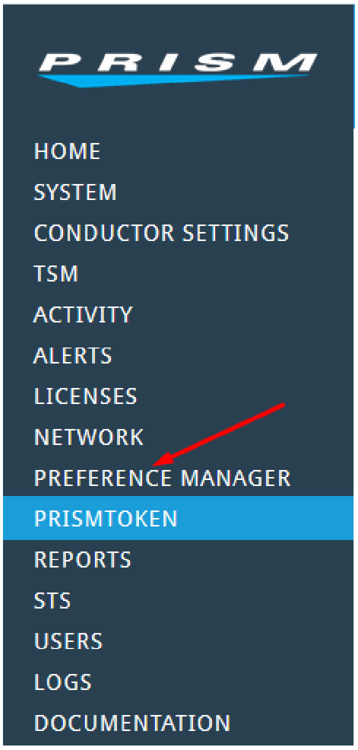
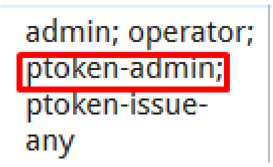


Figure 8 Preference Manager

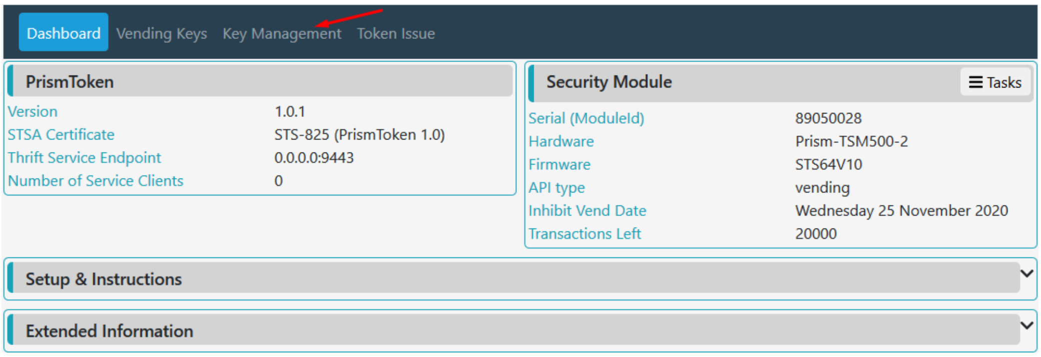
- Find the preference "ptoken.allowUnapprovedKmcPubkeys" and change it to "True".
- Next, login to TsmWeb as a PrismToken administrator. Note that a PrismToken administrator will have the role of 'ptoken-admin'.



admin; operator; ptoken-admin; ptoken-issue-any

Figure 9 ptoken Admin

- Click on "PrismToken" tab in the left navigation bar.
- Click the "Key Management" tab.



PrismToken

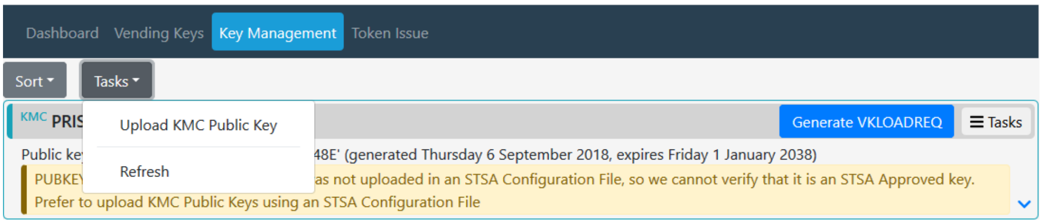
| | |
|---|---|
| Version | 1.0.1 |
| STSA Certificate | STS-825 (PrismToken 1.0) |
| Thrift Service Endpoint | 0.0.0.0:9443 |
| Number of Service Clients | 0 |

Security Module

| | |
|---|---|
| Serial (ModuleId) | 89050028 |
| Hardware | Prism-TSM500-2 |
| Firmware | STS64V10 |
| API type | vending |
| Inhibit Vend Date | Wednesday 25 November 2020 |
| Transactions Left | 20000 |

**Figure 10 Key Management tab**

- Click the "Tasks" dropdown at the top of the tab and select "Upload public key".



KMC PRIS

Public ke 48E' (generated Thursday 6 September 2018, expires Friday 1 January 2038)

PUBKEY as not uploaded in an STSA Configuration File, so we cannot verify that it is an STSA Approved key.
Prefer to upload KMC Public Keys using an STSA Configuration File

**Figure 11 Upload KMC Task**

- Open/view the PUBLIC KEY (plain text file) in a text editor then highlight and copy the entire contents.
- In the "Upload KMC Public Key" popup, paste the contents in the box labelled "Paste KMC PUBKEY", then click "Upload".

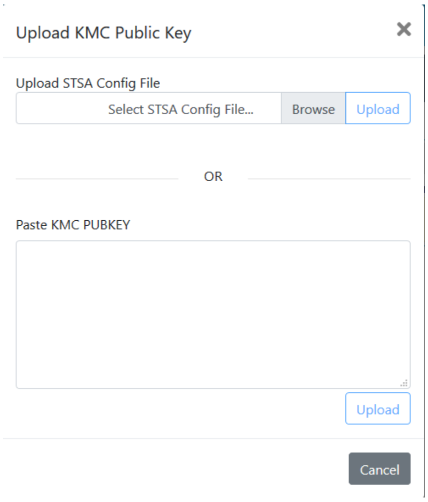


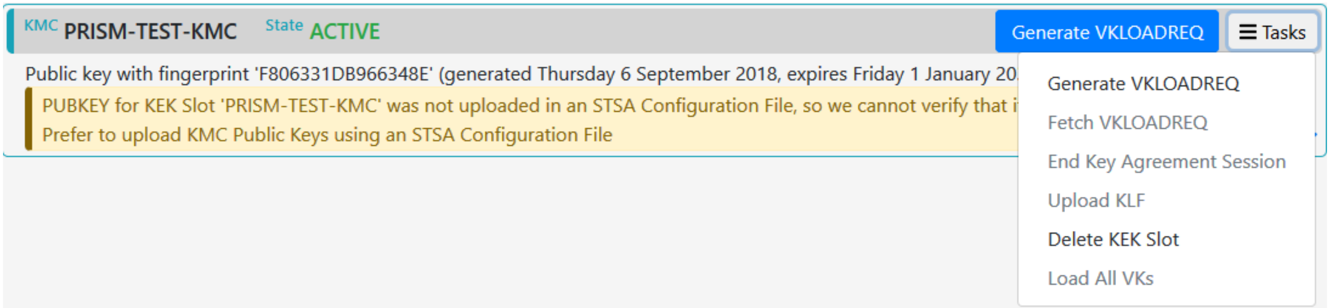
**Figure 12 Upload KMC Public Key**

## 5.6 Vending Keys

Firstly, ensure that the SGC owner has given the KMC a Key Use Authorisation allowing the SGC’s Vending Keys to be loaded into your Security Module.

### 5.6.1 Generate VKLOADREQ and upload KLF

In PrismToken, on the Key Management tab you should see a list of known Key Management Centres, if you don't you must load the KMC PUBKEY as explained in the previous step. Choose the KMC from which you need to obtain Vending Keys, click that KMC's "Tasks" menu and choose "Generate VKLOADREQ". Follow the on-screen instructions. You will need to copy & paste the VKLOADREQ into an e-mail and send it to the KMS via email (KMS Operator) for processing.



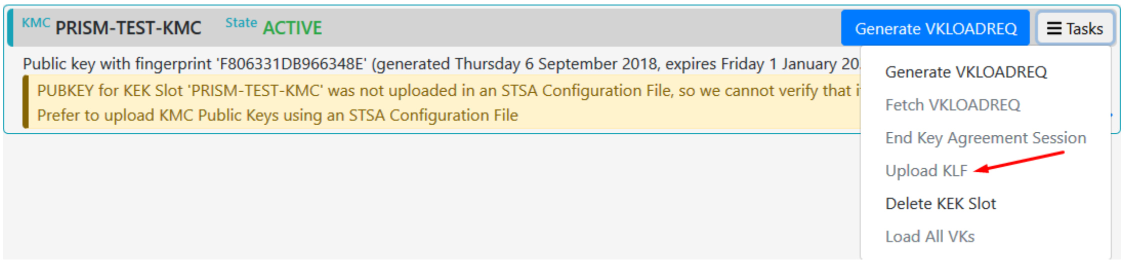
KMC PRISM-TEST-KMC State ACTIVE

Public key with fingerprint 'F806331DB966348E' (generated Thursday 6 September 2018, expires Friday 1 January 20

PUBKEY for KEK Slot 'PRISM-TEST-KMC' was not uploaded in an STSA Configuration File, so we cannot verify that i
Prefer to upload KMC Public Keys using an STSA Configuration File

Figure 13 Obtain Vending Keys

The KMC will reply with an e-mail that has a Key Load File attached. To upload the Key Load File go to PrismToken, Key Management tab, the KMC's "Tasks" menu, and choose "Upload KLF". You can then navigate to PrismToken's "Vending Keys" tab where you will see your vending keys.

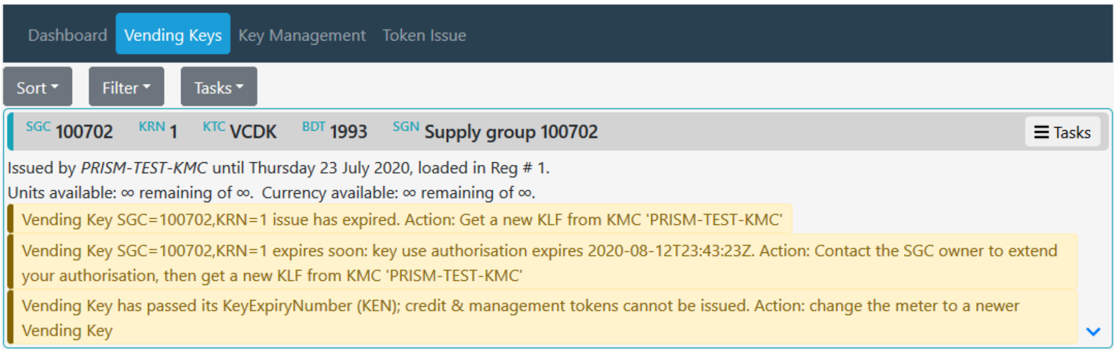


KMC PRISM-TEST-KMC State ACTIVE

Public key with fingerprint 'F806331DB966348E' (generated Thursday 6 September 2018, expires Friday 1 January 20

PUBKEY for KEK Slot 'PRISM-TEST-KMC' was not uploaded in an STSA Configuration File, so we cannot verify that i
Prefer to upload KMC Public Keys using an STSA Configuration File

Figure 14 Upload KLF



SGC 100702 KRN 1 KTC VCDK BDT 1993 SGN Supply group 100702

Issued by *PRISM-TEST-KMC* until Thursday 23 July 2020, loaded in Reg # 1.
Units available: ∞ remaining of ∞. Currency available: ∞ remaining of ∞.

Vending Key SGC=100702,KRN=1 issue has expired. Action: Get a new KLF from KMC 'PRISM-TEST-KMC'

Vending Key SGC=100702,KRN=1 expires soon: key use authorisation expires 2020-08-12T23:43:23Z. Action: Contact the SGC owner to extend your authorisation, then get a new KLF from KMC 'PRISM-TEST-KMC'

Vending Key has passed its KeyExpiryNumber (KEN); credit & management tokens cannot be issued. Action: change the meter to a newer Vending Key

Figure 15 Vending Keys

### 5.6.2 Load All VKs

Uploading the KLF loads the information from the KLF into PrismToken and it passes the VKLOADRSP to the security module. PrismToken does not load any of the keys from the KLF into the security module. When the security module processes the VKLOADRSP the vending keys for that KEK slot are erased from the security module.

To load the keys from the KLF you will need to select the “Load All VKs” option from the “Tasks” menu. Take note that if the key agreement session is ended before loading all the vending keys then it will not be possible to load vending keys from the same KLF file into the security module. A new VKLOADREQ will need to be generated and sent to the KMS via email (KMS Operator) for processing. The KLF returned from the KMS needs to be uploaded and then you will need to select the “Load All VKs” option from the “Tasks” menu. Refer to Section 5.6.3 if “Load All VKs” returns the following error Failed to load all VKs: STS API error VK_REG_NOT_FOUND_ERROR (code 28)

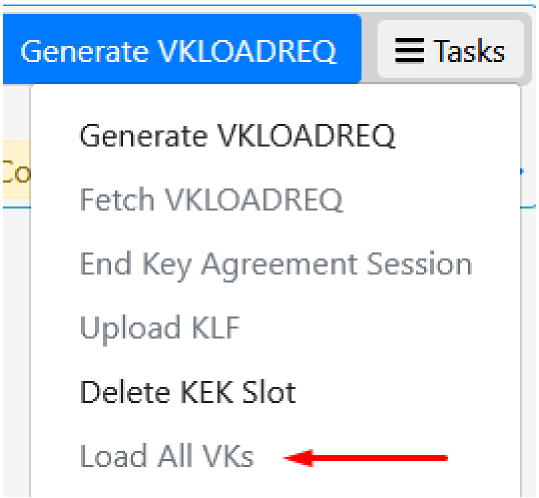


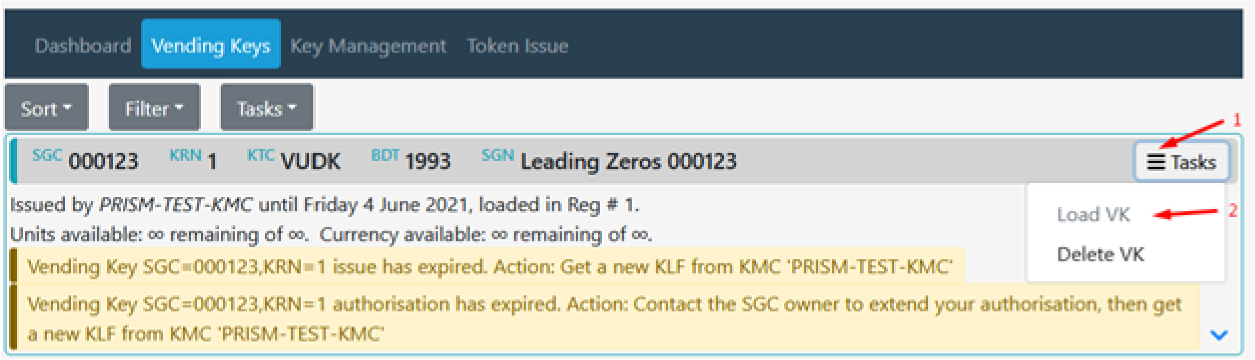
**Figure 16 Load All VKs**

### 5.6.3 Load Individual VKs

There may be circumstances where the number of vending keys within a security module’s KLF exceeds the maximum number of vending key registers available in the security module. This is a hardware limit based on the TSM family. A TSM250 has a limit of 25 key registers/slots whereas the TSM500i supports up to 999. Exceeding this limit is often due to TSM250 customers requesting far too many of the manufacturer default SGC’s. In practical terms, very few of these default SGC’s are required. The STSA have introduced two universal default Supply Group Codes (SGC) which are available to anyone:

- 1993 Base Date Universal Default 991993
- 2014 Base Date Universal Default 999014

After uploading the KLF on the Key Management tab, instead of using the “Load all Vending Keys” button, navigate to the Vending Keys tab and change the “Filter” to Show all Vending Keys and not just those in the security module. There is a task button per vending key, where there are two options. You can load or delete the vending key from the security module. The key agreement session must be ended after loading the individual vending keys to complete the process.



SGC 000123 KRN 1 KTC VUDK BDT 1993 SGN Leading Zeros 000123

Issued by *PRISM-TEST-KMC* until Friday 4 June 2021, loaded in Reg # 1.
Units available: ∞ remaining of ∞. Currency available: ∞ remaining of ∞.

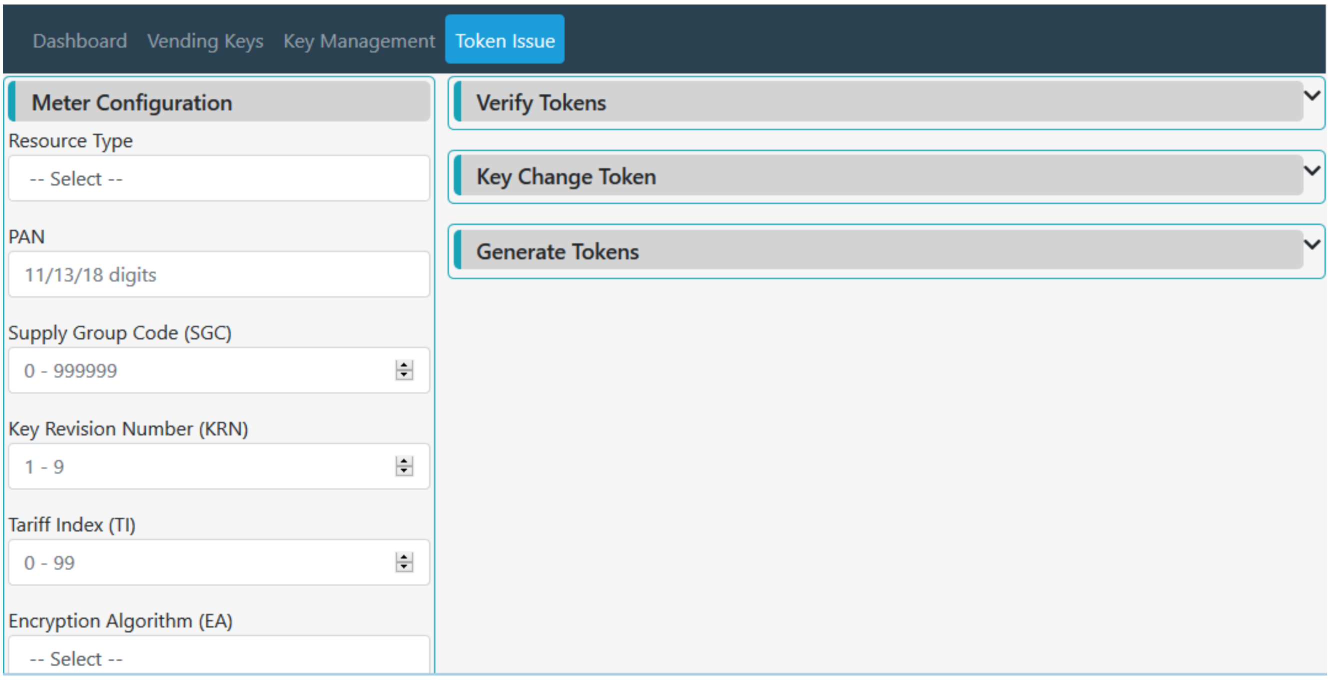
Vending Key SGC=000123,KRN=1 issue has expired. Action: Get a new KLF from KMC 'PRISM-TEST-KMC'

Vending Key SGC=000123,KRN=1 authorisation has expired. Action: Contact the SGC owner to extend your authorisation, then get a new KLF from KMC 'PRISM-TEST-KMC'

Figure 17 Load or delete VK

## 5.7 Token Issue

Click on “PrismToken” in the left navigation bar and then select the “Token Issue” tab. Enter your meter configuration on the left and use the boxes on the right to choose the type of token you want.



Meter Configuration

Resource Type: -- Select --

PAN: 11/13/18 digits

Supply Group Code (SGC): 0 - 999999

Key Revision Number (KRN): 1 - 9

Tariff Index (TI): 0 - 99

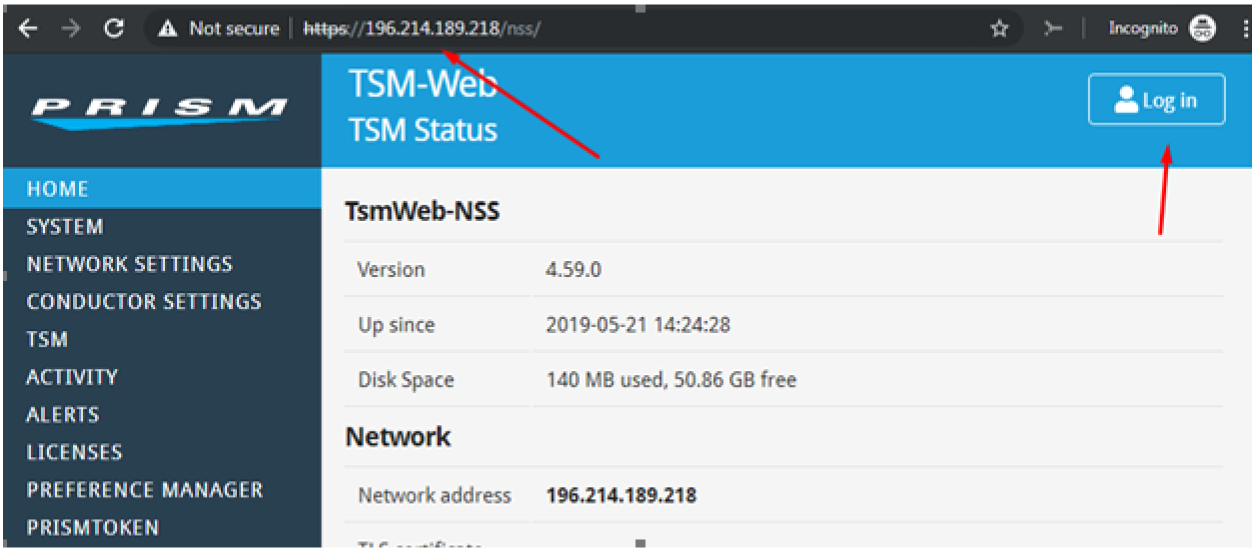
Encryption Algorithm (EA): -- Select --

Figure 18 Token Issue

### 5.7.1 Generating STS tokens from the “Token Issue” tab

The “Token Issue” tab is a PrismToken User Interface that can be used by technicians to generate engineering and key change tokens for a meter. Our demo installation is also set up to allow credit tokens to be generated although credit token generation via this user interface would be disabled in a live system.

- Ensure the URL for the PrismToken Web UI into your browser’s address bar.
- On the home page, click on the “Log in” button.



TsmWeb-NSS

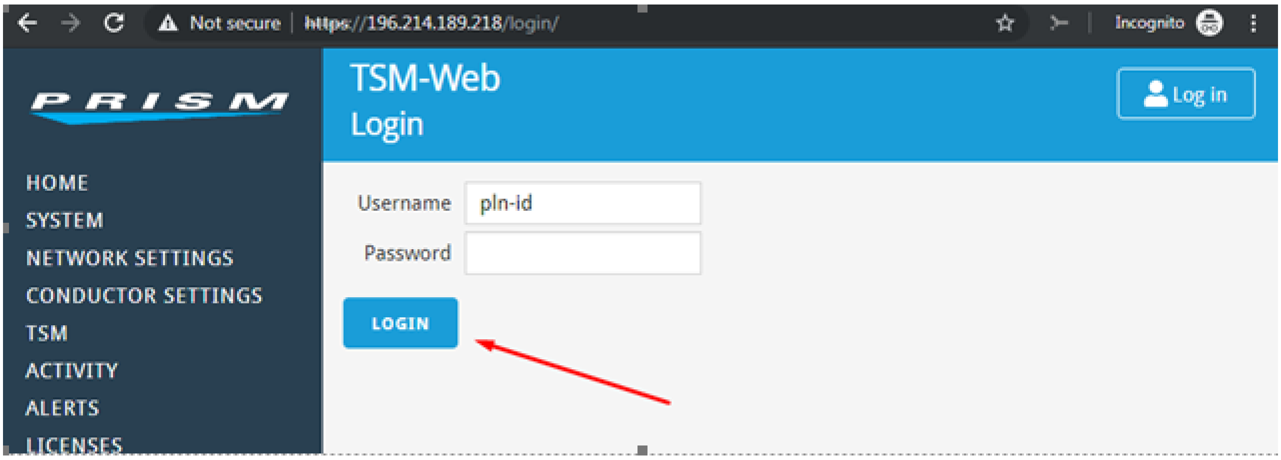
| | |
|---|---|
| Version | 4.59.0 |
| Up since | 2019-05-21 14:24:28 |
| Disk Space | 140 MB used, 50.86 GB free |

Network

| | |
|---|---|
| Network address | **196.214.189.218** |

**Figure 19 STS Token Home**

- Log in by entering the username and password supplied in the e-mail chain below and clicking the "LOGIN" button.

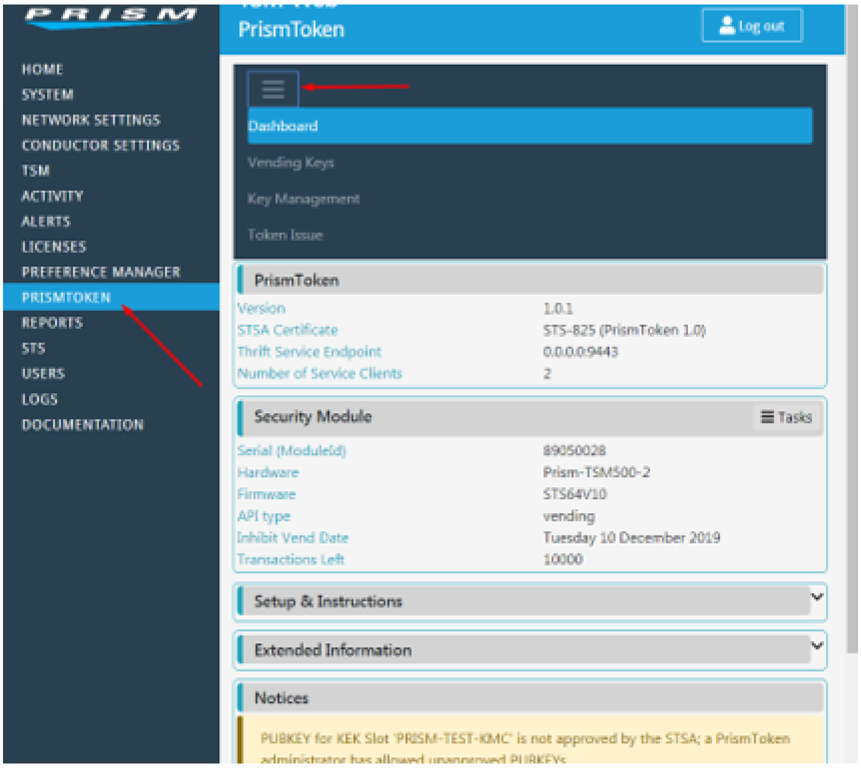


Username: pln-id

Password:

**Figure 20 STS Token Login**

- On the left navigation menu click "PRISMTOKEN", this will take you to the PrismToken Dashboard.
- If you are using a device with a narrow screen you may need to click the "Hamburger" menu icon to see the other PrismToken tabs.



PrismToken

|  |  |
|---|---|
| Version | 1.0.1 |
| STSA Certificate | STS-825 (PrismToken 1.0) |
| Thrift Service Endpoint | 0.0.0.0:9443 |
| Number of Service Clients | 2 |

Security Module

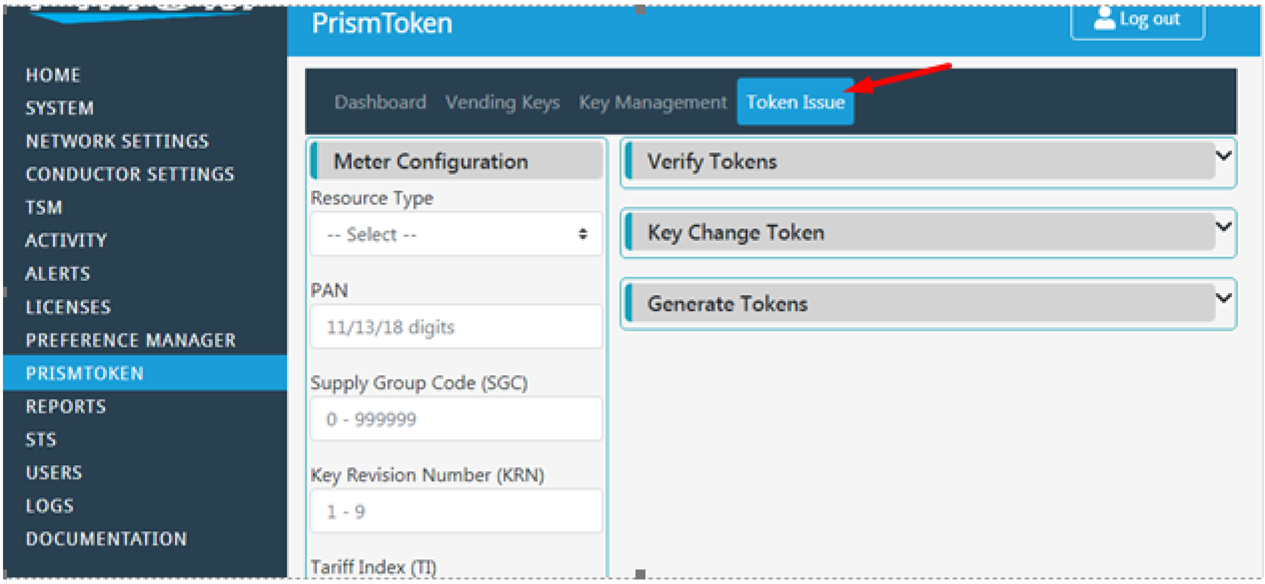
|  |  |
|---|---|
| Serial (ModuleId) | 89050028 |
| Hardware | Prism-TSM500-2 |
| Firmware | STS64V10 |
| API type | vending |
| Inhibit Vend Date | Tuesday 10 December 2019 |
| Transactions Left | 10000 |

Notices

PUBKEY for KEK Slot 'PRISM-TEST-KMC' is not approved by the STSA; a PrismToken administrator has allowed unapproved PUBKEYs

**Figure 21 STS Token PrismToken Home**

- Click the “Token Issue” tab



Meter Configuration

Resource Type: -- Select --

PAN: 11/13/18 digits

Supply Group Code (SGC): 0 - 999999

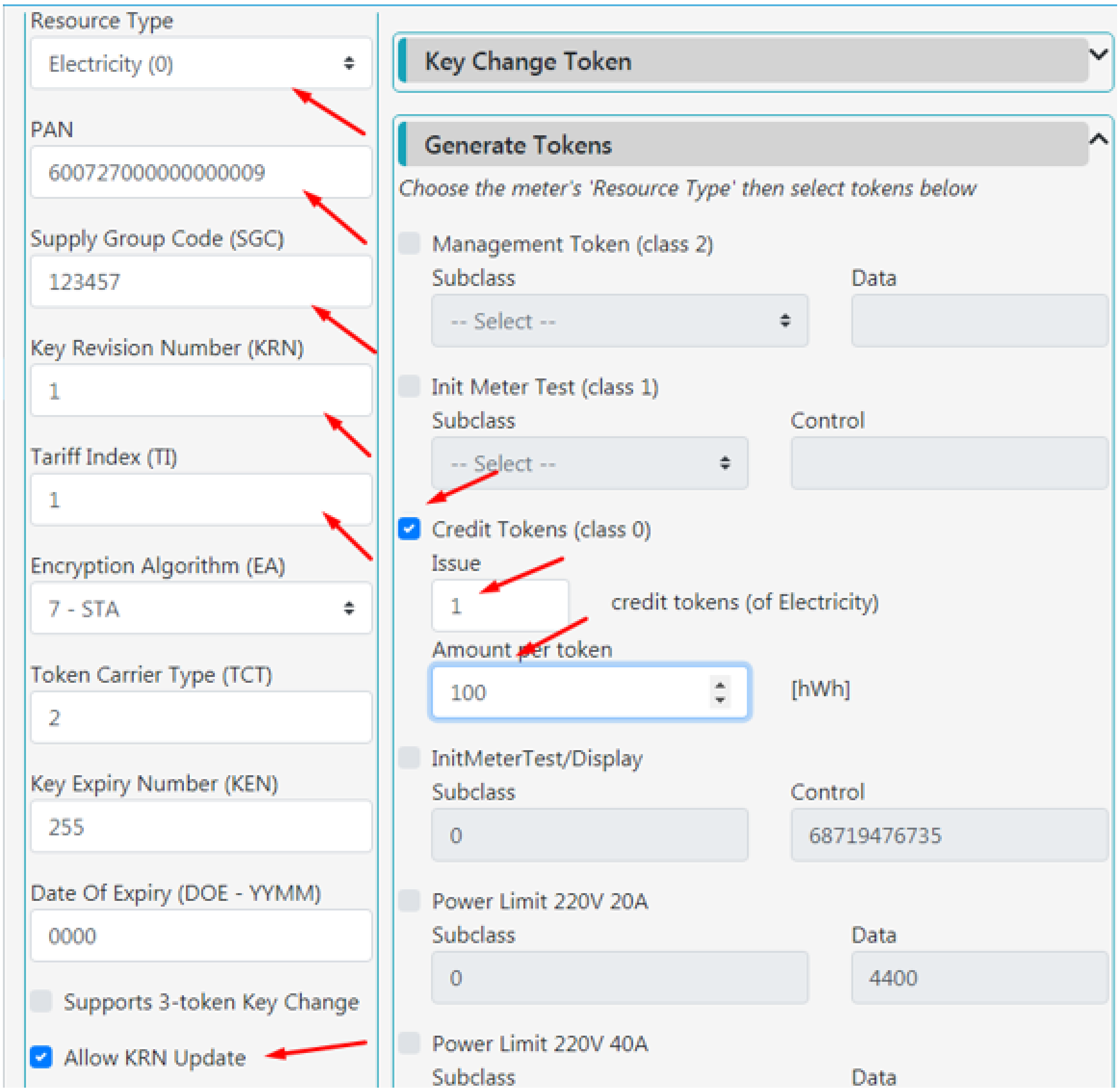
Key Revision Number (KRN): 1 - 9

Tariff Index (TI)

**Figure 22 STS Token Issue Button**

- You can now enter the configuration information for a meter, and the parameters for a token, then generate the token.
- The example below illustrates generating a single 100 hWh credit token for the electricity meter with PAN 600727000000000009, SGC 123457, KRN 1, TI 1. By ticking the “Allow KRN Update” box will tell PrismToken that I also want to receive Key Change Tokens to move the

meter to the latest available KRN. Once the data is entered, I clicked the "Generate Tokens" button at the bottom of the page to generate the tokens.



Resource Type
Electricity (0)

PAN
600727000000000009

Supply Group Code (SGC)
123457

Key Revision Number (KRN)
1

Tariff Index (TI)
1

Encryption Algorithm (EA)
7 - STA

Token Carrier Type (TCT)
2

Key Expiry Number (KEN)
255

Date Of Expiry (DOE - YYMM)
0000

□ Supports 3-token Key Change

☑ Allow KRN Update

*Choose the meter's 'Resource Type' then select tokens below*

□ Management Token (class 2)
Subclass: -- Select --
Data:

□ Init Meter Test (class 1)
Subclass: -- Select --
Control:

☑ Credit Tokens (class 0)
Issue 1 credit tokens (of Electricity)
Amount per token 100 [hWh]

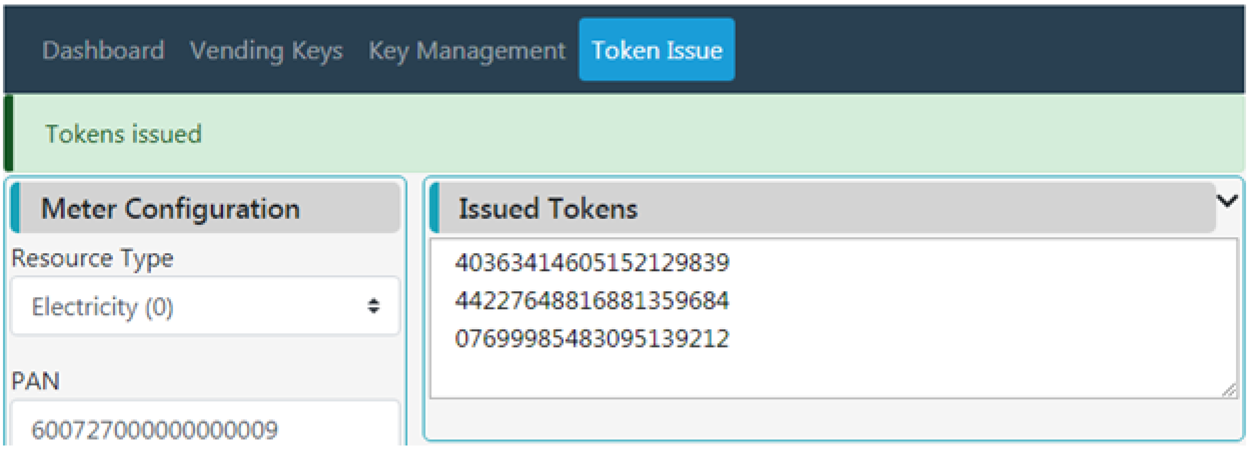
□ InitMeterTest/Display
Subclass: 0
Control: 68719476735

□ Power Limit 220V 20A
Subclass: 0
Data: 4400

□ Power Limit 220V 40A
Subclass
Data

**Figure 23 STS Token example**

- The results are as follows: the system has generated three tokens (2 Key Change Tokens, plus one Credit Token). These tokens will work in meter 600727000000000009 if it has been configured on SGC 122457, KRN 1, TI 1.



Tokens issued

Meter Configuration

Resource Type

Electricity (0)

PAN

600727000000000009

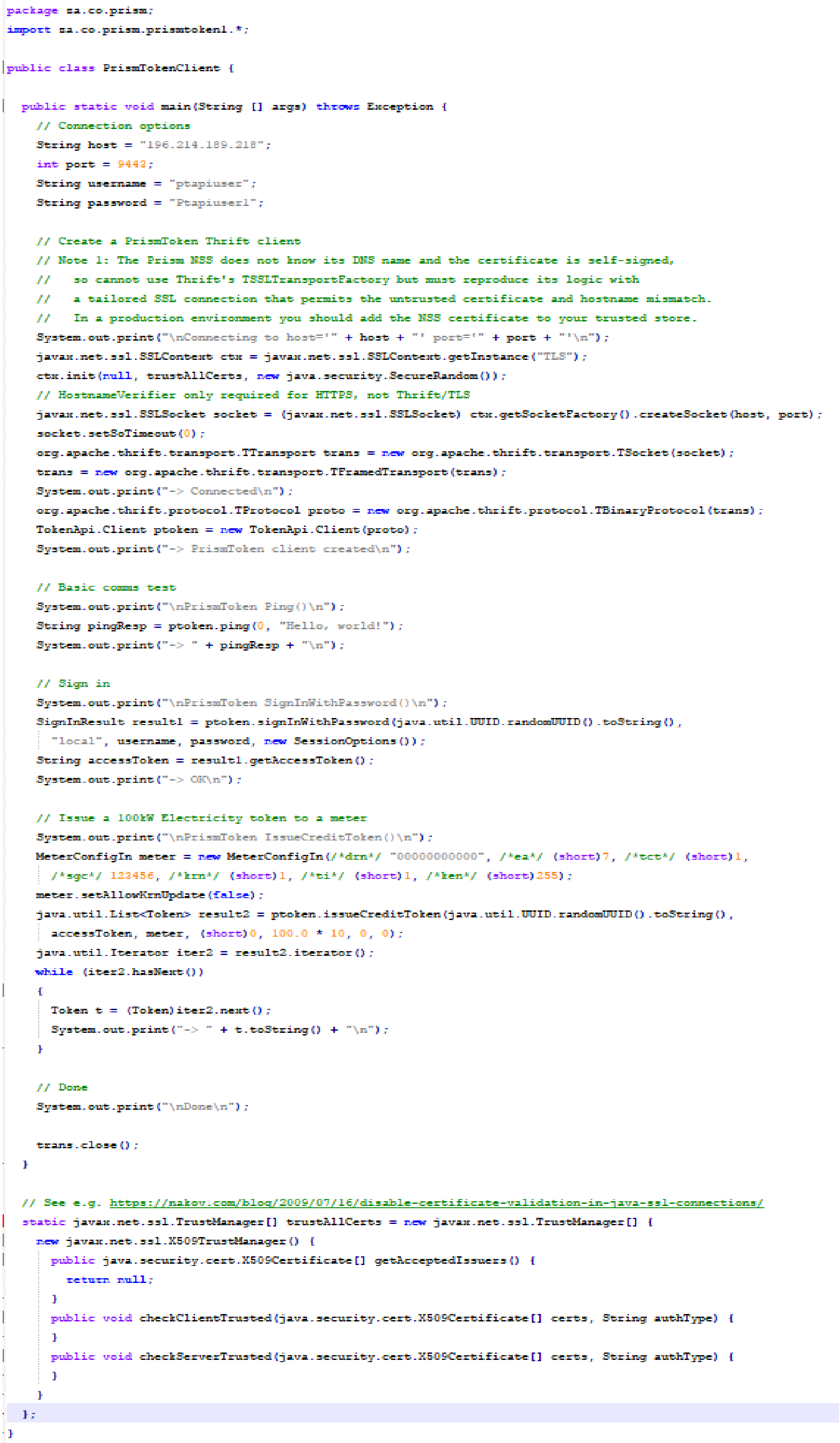
Issued Tokens

40363414605152129839
44227648816881359684
07699985483095139212

Figure 24 STS Token generated Tokens

### 5.7.2 Generating STS tokens by integrating with the Thrift API

- In a typical STS deployment, a Vending System would manage a database of meters with associated point of connection, tariff, customer, etc. The Vending System would integrate with PrismToken to generate tokens (much like legacy Vending Systems integrate with the TSM210 or TSM410 Security Module).
- The documents "Guide to Using Thrift" and "PrismToken Thrift API" (both attached) explain the Thrift protocol and the functions available in the PrismToken API (which is accessed using the Thrift protocol).
- You will need to develop client software that calls the PrismToken Thrift API to generate the Key Change Tokens and Credit Tokens as they are needed by the Vending System.
    a) Your client software will call the issueCreditToken() function to issue a Credit token (optionally with Key Change Tokens to update the meter to the latest KRN).
    b) Your client software may also call the issueKeyChangeTokens() function to issue Key Change Tokens explicitly.
- Below is a code snippet that comes from the Java development pack for PrismToken and is a simple example of how client software uses the Thrift library to call PrismToken.



```java
package sa.co.prism;
import sa.co.prism.prismtoken1.*;

public class PrismTokenClient {

  public static void main(String [] args) throws Exception {
    // Connection options
    String host = "196.214.189.218";
    int port = 9443;
    String username = "ptapiuser";
    String password = "Ptapiuser1";

    // Create a PrismToken Thrift client
    // Note 1: The Prism NSS does not know its DNS name and the certificate is self-signed,
    //   so cannot use Thrift's TSSLTransportFactory but must reproduce its logic with
    //   a tailored SSL connection that permits the untrusted certificate and hostname mismatch.
    //   In a production environment you should add the NSS certificate to your trusted store.
    System.out.print("\nConnecting to host='" + host + "' port='" + port + "'\n");
    javax.net.ssl.SSLContext ctx = javax.net.ssl.SSLContext.getInstance("TLS");
    ctx.init(null, trustAllCerts, new java.security.SecureRandom());
    // HostnameVerifier only required for HTTPS, not Thrift/TLS
    javax.net.ssl.SSLSocket socket = (javax.net.ssl.SSLSocket) ctx.getSocketFactory().createSocket(host, port);
    socket.setSoTimeout(0);
    org.apache.thrift.transport.TTransport trans = new org.apache.thrift.transport.TSocket(socket);
    trans = new org.apache.thrift.transport.TFramedTransport(trans);
    System.out.print("-> Connected\n");
    org.apache.thrift.protocol.TProtocol proto = new org.apache.thrift.protocol.TBinaryProtocol(trans);
    TokenApi.Client ptoken = new TokenApi.Client(proto);
    System.out.print("-> PrismToken client created\n");

    // Basic comms test
    System.out.print("\nPrismToken Ping()\n");
    String pingResp = ptoken.ping(0, "Hello, world!");
    System.out.print("-> " + pingResp + "\n");

    // Sign in
    System.out.print("\nPrismToken SignInWithPassword()\n");
    SignInResult result1 = ptoken.signInWithPassword(java.util.UUID.randomUUID().toString(),
      "local", username, password, new SessionOptions());
    String accessToken = result1.getAccessToken();
    System.out.print("-> OK\n");

    // Issue a 100kW Electricity token to a meter
    System.out.print("\nPrismToken IssueCreditToken()\n");
    MeterConfigIn meter = new MeterConfigIn(/*drn*/ "00000000000", /*ea*/ (short)7, /*tct*/ (short)1,
      /*sgc*/ 123456, /*krn*/ (short)1, /*ti*/ (short)1, /*ken*/ (short)255);
    meter.setAllowKrnUpdate(false);
    java.util.List<Token> result2 = ptoken.issueCreditToken(java.util.UUID.randomUUID().toString(),
      accessToken, meter, (short)0, 100.0 * 10, 0, 0);
    java.util.Iterator iter2 = result2.iterator();
    while (iter2.hasNext())
    {
      Token t = (Token)iter2.next();
      System.out.print("-> " + t.toString() + "\n");
    }

    // Done
    System.out.print("\nDone\n");

    trans.close();
  }

  // See e.g. https://nakov.com/blog/2009/07/16/disable-certificate-validation-in-java-ssl-connections/
  static javax.net.ssl.TrustManager[] trustAllCerts = new javax.net.ssl.TrustManager[] {
    new javax.net.ssl.X509TrustManager() {
      public java.security.cert.X509Certificate[] getAcceptedIssuers() {
        return null;
      }
      public void checkClientTrusted(java.security.cert.X509Certificate[] certs, String authType) {
      }
      public void checkServerTrusted(java.security.cert.X509Certificate[] certs, String authType) {
      }
    }
  };
}
```

**Figure 25 Thrift Java example**

- Currently PrismToken has development packs for C#, Java and PHP. Other language libraries can be made available on request.

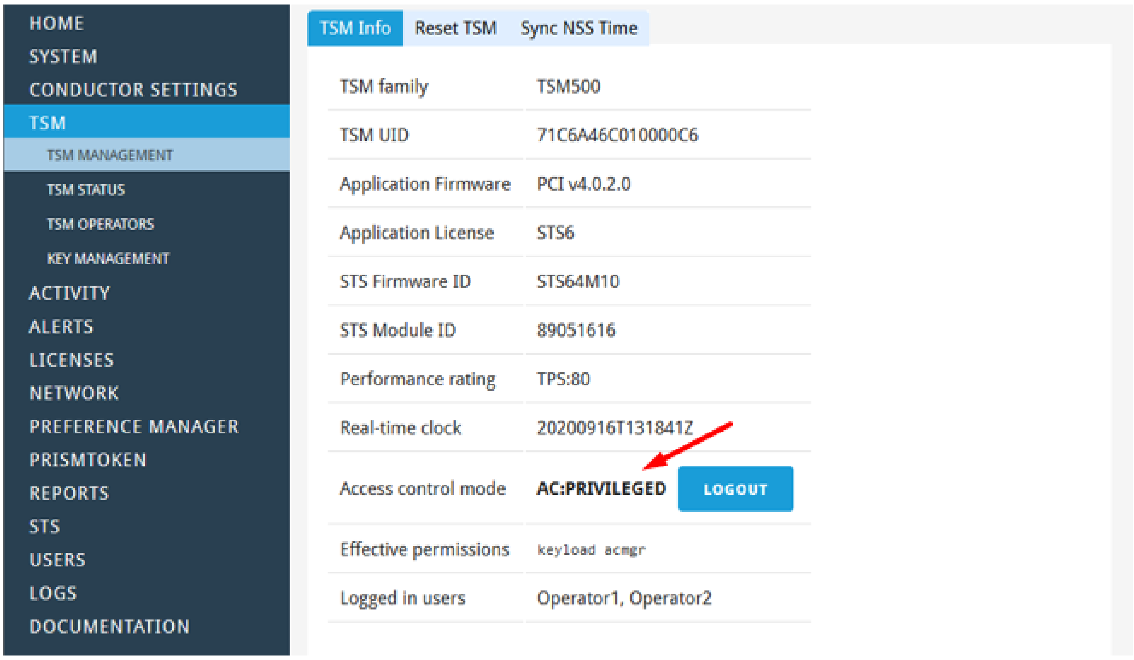
# 6 DITK

The DITK is also known as the Dispenser ROM key. The PrismToken for Manufacturing needs to have at least one DITK loaded to allow newly manufactured meters to be key changed from the DITK. The DITK is entered in the form of components using a Key Component Entry Device (KCED) connected to serial port. If you choose to use your existing DITK then there is no need to generate one.

**It is important to note that the DITK loaded into the HSM must match the actual DITK used by the meter!!**

## 6.1 Load DITK

- Connect a KCED to the NSS's "CSP" port which is located on the front of the TSM500i-NSS.
- Set the Security Module into PRIVILEGED state.
  a) Click on the "TSM" tab in the left navigation menu, then select the "TSM Info" tab.
  b) If the page has a warning that the HSM is in Boot Loader mode, then you must reboot the Security Module by selecting the "Reset TSM" tab and clicking "Reset to App". Also refer to section 2.8.5 Put the TSM500i into the Application State in TSM500i and TsmWeb User Guide (PR-D2-0854).
  c) The "Access control mode" should be "AC:OPERATIONAL", if this isn't the case you should contact support.
  d) On the "TSM Info" tab click on the "Login Operator" button, follow the prompts on the screen and on the KCED. You will be instructed to enter your Crypto Officer passwords.
  e) The "Access control mode" should now be "AC:PRIVILEGED".
  f) The STS6 application firmware will exit the privileged state automatically if any one of the following conditions are met:
     i. After 30 minutes the HSM will return to operational state.
     ii. After 15 minutes where no calls have been made to the HSM.
     iii. if more than 50 calls are made then the HSM will return to operational state.
     iv. Note that every time you navigate to a different page on TSMWeb this will count towards a call. Some pages require more than one call. Therefore, this may result in the HSM returning to operational state if more than 50 calls are made. Its advised that once you are in Privileged state you should proceed to load DITK immediately.



| | |
|---|---|
| TSM family | TSM500 |
| TSM UID | 71C6A46C010000C6 |
| Application Firmware | PCI v4.0.2.0 |
| Application License | STS6 |
| STS Firmware ID | STS64M10 |
| STS Module ID | 89051616 |
| Performance rating | TPS:80 |
| Real-time clock | 20200916T131841Z |
| Access control mode | AC:PRIVILEGED |
| Effective permissions | `keyload acmgr` |
| Logged in users | Operator1, Operator2 |

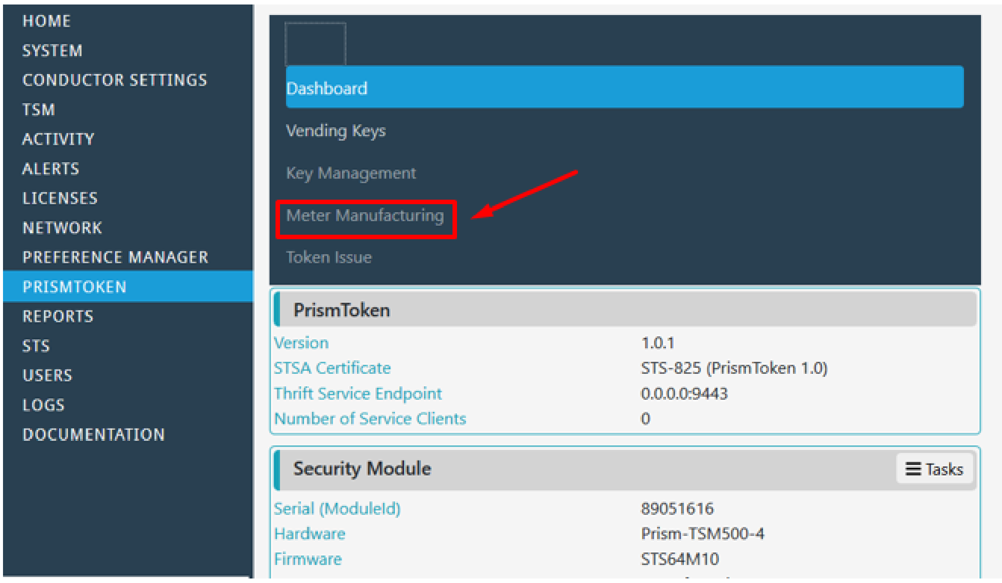
**Figure 26 HSM state**

## 6.1.1 Generate the DITK

There are separate DITKs for EA=07 which is the STS algorithm and EA=11 which is the MISTY1 algorithm. If you do not manufacture EA=11 meters then you will only require an EA=07 DITK. Each DITK requires two or three components and a Key Check Value (KCV). You can generate the DITK using PrismToken or use a pre-existing DITK. If you choose to use a pre-existing DITK that was not generated using PrismToken please refer to section “5.2 DITK is known”. The DITK must be generated and stored in the form of components, which are split between two or three trusted custodians. The key components are generated such that combining all components using XOR will yield the DITK.

### 6.1.1.1 *To generate a DITK:*

- Ensure that the module is in a privileged state as mentioned at the beginning of this section.
- Click on the “PrismToken” left in the left navigation menu.
- Select the “Meter Manufacturing” tab. Click “Generate & Display Key Components” to expand the box.



PrismToken

| | |
|---|---|
| Version | 1.0.1 |
| STSA Certificate | STS-825 (PrismToken 1.0) |
| Thrift Service Endpoint | 0.0.0.0:9443 |
| Number of Service Clients | 0 |

Security Module

| | |
|---|---|
| Serial (ModuleId) | 89051616 |
| Hardware | Prism-TSM500-4 |
| Firmware | STS64M10 |

Figure 27 Meter Manufacturing

- Select the EA and the number of components (equal to the number of custodians who will look after the key), then click “View on KCED”.

  **It is important to note that generating an EA7 DITK using the PrismToken UI will generate an ODD parity DITK**.

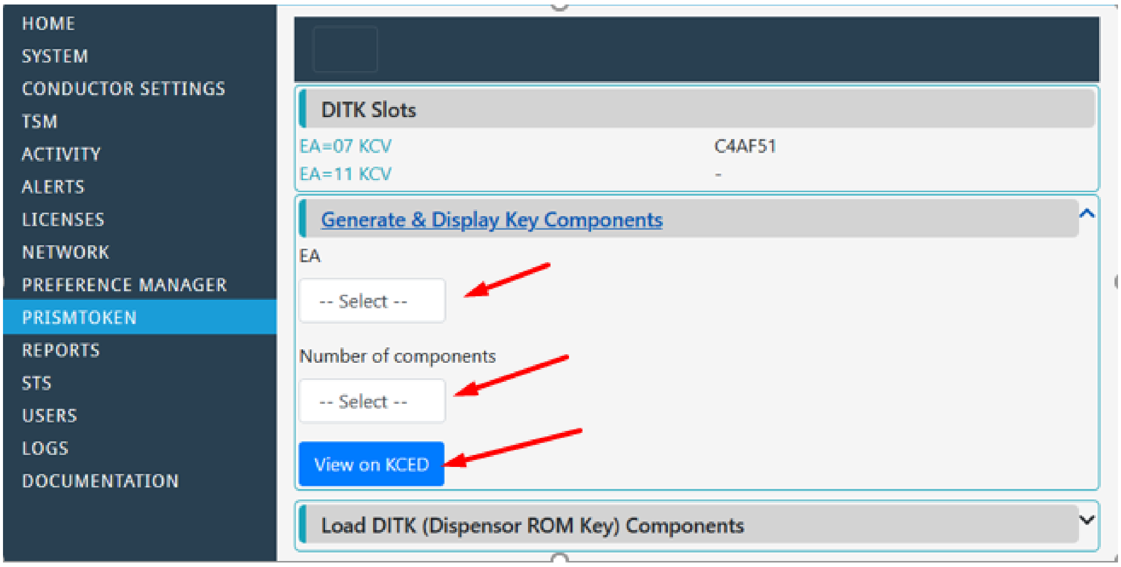
  **This is consistent with our recommendation that the EA7 DITK is generated with ODD parity, as the check digits (KCV) ignore the parity bits.**

- Each key custodian must follow the instructions on the KCED, write down their component and the KCED, and keep the written component stored in a safe place. For convenience a DITK component sheet template is provided in section 6.1.2 below.

  You will need these components whenever you need to load the DITK into the Security Module (typically immediately and in case of disaster recovery).

- You will need to repeat this process if you want to add another EA.

[i] ***Prism Support can find a history of DITK odd-parity problems at WI314***



DITK Slots

EA=07 KCV C4AF51

EA=11 KCV -

EA

-- Select --

Number of components

-- Select --

Figure 28 Generate DITK

### 6.1.2 DITK Component Sheet

Key name ______________________________

IDENTIFYING NAME OR DESCRIPTION OF THE KEY

Date ______________________________

YYYY/MM/DD OF KEY / COMPONENT CREATION

Generated by ______________________________

FULL NAME AND CONTACT NUMBER OF CUSTODIAN (AT KEY GENERATION)

Component number ______ *of* _______ Algorithm (circle one right): EA7 / EA11

Component *Fill in those parts that are applicable.*

Part 1: *(For EA7 and EA11 DITK components)*

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | |

Part 2: *(Only for EA11 DITK key components)*

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | |

Component Check value

| | |
|---|---|
| □□□□□□<br>**Key Check Value**<br>□□□□□□ | Qwerty |

## 6.2 DITK is known

It is important to note:

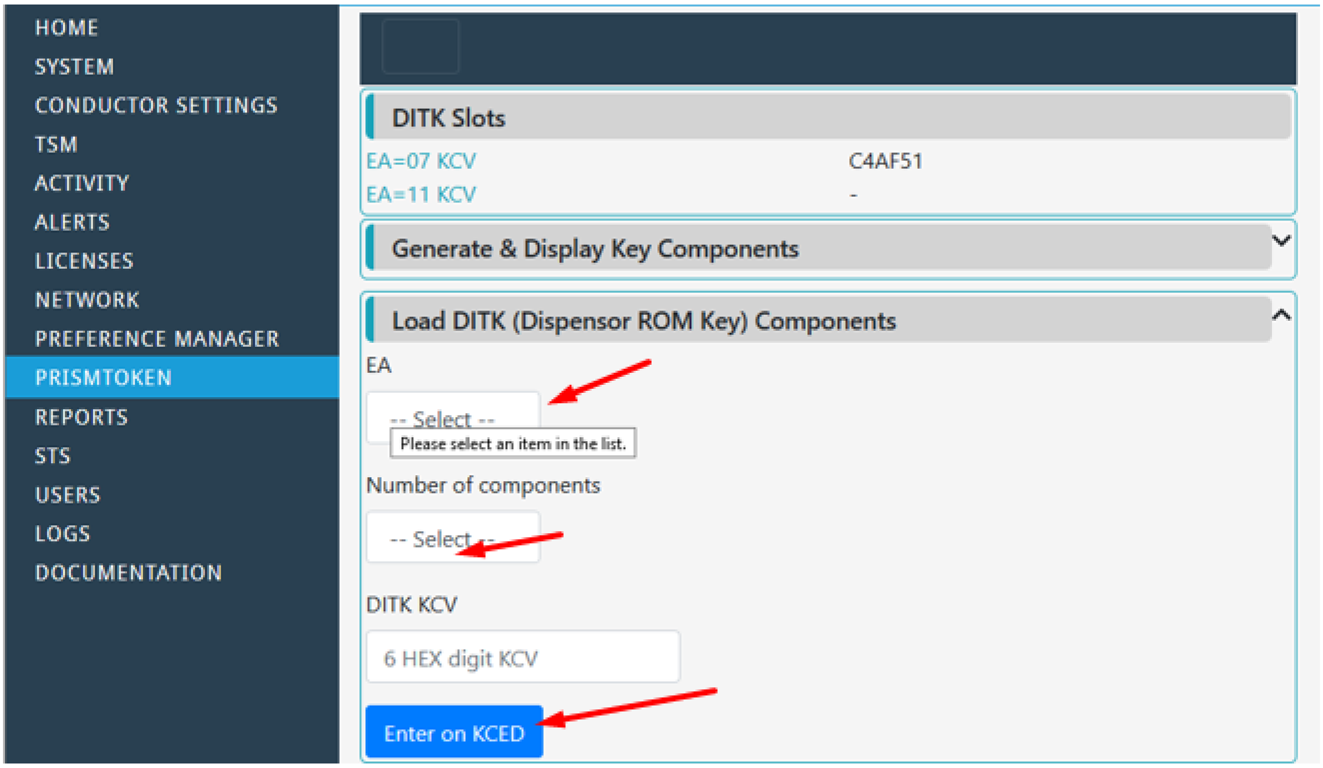
- The meter manufacturer must check whether the process of injecting the key into the meter, or possibly the meter itself, modifies the DITK in any way before using it (e.g. by setting parity to ODD), and ensure that the DITK loaded into the SM must match the actual DITK used by the meter.
- We would recommend using the STS Simulator (available from the STS Association) to generate a "Clear Tamper" token under the DITK that you think is being used by the meter. If the token is accepted by the meter then the DITK is correct.
- Some meter manufacturers who used STS/Legacy firmware loaded their DITK using the SM?IK with (S)et Parity option. They in particular must take care to understand what actual DITK value is used by the meter.
- The DITK loaded into the HSM must match the actual DITK used by the meter.

If your DITK is already known, you must assign components such that if you combine all the components using XOR it will yield the DITK. The DITK could be split between two or three custodians. If you choose to split your DITK between two custodians then you will require two components, alternatively if you choose to split your DITK between three custodians you will require three components.

For example if your actual EA7 DITK used by the meter is x'7676767676767676 (KCV = x'4CBE91) you could choose to split it between two or three components therefore having two or three custodians.

Below is an example of the DITK being split amongst 2 components.

| | |
|---|---|
| DITK Component 1 | x'0000000076767676 |
| DITK Component 2 | x'7676767600000000 |
| DITK Component 1 XORED DITK Component 2 | x'7676767676767676 (DITK) |



DITK Slots

| | |
|---|---|
| EA=07 KCV | C4AF51 |
| EA=11 KCV | - |

EA

-- Select --

Please select an item in the list.

Number of components

-- Select --

DITK KCV

6 HEX digit KCV

**Figure 29 DITK is known**

**It is important to note that with reference to the EA7 DITK, the KCV calculation ignores the parity bits, and that if the meter and the HSM are using keys that differ only in parity bits the KCV will be the same but the crypto operations will fail !!**

# 7 Automating PrismToken using the Thrift API

PrismToken has a remotely callable Thrift API that can be used to integrate its token issue capabilities into your applications. See the documents PR-D2-0984 “Guide to Using Thrift” and PR-D2-1009 “PrismToken Thrift API” to learn more about Thrift and the API. Client code for the Thrift API is generated using the open source software “Apache Thrift” and the API definition file “prismtoken1-TokenApi.thrift” which you will find under TsmWeb Documentation. We can supply pre-built client code with examples for the C#, Java, and PHP languages. If you are using another language, please contact us for advice. The PrismToken Thrift API service is accessible at your NSS’s IP address, on port 9443 by default (you can check the port number on the PrismToken dashboard). A client must authenticate itself by calling the signInWithPassword() method using realm=”local” and the credentials of a TsmWeb user account that has ‘ptoken-api’ role (plus ‘ptoken-issue-*’ roles as required).

# 8 Operations and Monitoring

This section provides guidance for operating PrismToken securely and reliably in a Live environment.

## 8.1 General

General guidance for Live environments:

- IT equipment can fail unexpectedly. You should always have a Disaster Recovery (DR) plan. That plan should include a standby PrismToken and Security Module, fully loaded with Vending Keys, and monitored regularly to ensure that the standby system is functional (with non-expired keys!). You will also need to plan how to fail over to the standby system.

## 8.2 Security in a Live environment

- Do not expose PrismToken, TsmWeb, or the NSS to the Internet. Deploy PrismToken on a network segment that is firewalled from inbound Internet traffic. Use a packet filter or VPNs/tunnels to restrict access to only those clients/networks that need it.
- Follow the relevant setup guides for the NSS, TsmWeb, and the Security Module.
  - Once the administrator password is set the NSS is secure-by-default and no additional security configuration is required for a Live environment.
- Authentication to PrismToken and/or TsmWeb is by username and password. Use strong passwords on all accounts. Restrict accounts to the minimum roles required to fulfil the account's purpose. Disable unused accounts.
- Client software that integrates with the PrismToken API should verify the TLS certificate of the server (i.e. TsmWeb/PrismToken). You can obtain the TLS certificate from the Network page in TsmWeb, by using the "Tasks" button of the "SSL/TLS Server Properties" section; alternatively you can query the TLS certificate using a tool like openssl. Client authentication by means of client-side certificate is not supported.
  - Procedures for converting the certificate format (if necessary) and importing it into your client's trusted store vary by client, development language, and operating environment, and are beyond the scope of this document. Your client software should document the steps required.

## 8.3 VendingKey settings at the KMC

Various VendingKey attributes that are set at the KMC can influence your operational environment. If you own the Supply Group then you have control over these attributes.

- The VendingKey's Refresh Period determines how frequently keys must be refreshed, i.e. how frequently you must get a fresh Key Load File (KLF) from the KMC. On PrismToken the Vending Key's "IUT" (Issued Until) attribute indicates when the current issue will expire and prevent the key from being used. The IUT may be different for every Vending Key.
- The VendingKey's Key Expiry Number (KEN) and Expiry (EXP) determine when the Vending Key expires and *will no longer be issued by the KMC or allowed to be used in any Security Module*. The KEN and EXP are usually set to a far-future date, but in some cases (such as a compromised key) these dates may be brought forward to force a key change on all meters.

- The VendingKey's Subclass Bitmap (SBM) can restrict which resource subclasses (Electricity, Water, Gas, Time, Currency) are allowed in a Credit vend. SBM is usually set to 0x00FF or 0xFFFF to allow all subclasses.

## 8.4 PrismToken Setup

The following settings can accessed through the Preference Manager, and can help you monitor PrismToken:

- If you use PrismToken in Meter Manufacturing then ensure that *ptoken.meterManufacturingMode* is set.
- If you use PrismToken in Vending and need to issue proprietary Manufacturer tokens then ensure that *ptoken.allowIssueProprietaryTokens* is set.
- Setting *ptoken.sessionTimeoutSec* determines the duration (in seconds) of an authentication session with the PrismToken API. The client must re-authenticate when the session expires. Your client software should document an appropriate range for this setting. The default is 300 second, which we believe is a good trade-off between security and usability. We recommend against values outside the range 120-86400.
- Settings *ptoken.inhibitVendWarningDays* and *ptoken.txCounterLowThreshold* cause SM License warnings on the PrismToken dashboard when these thresholds are crossed.
- Settings *ptoken.keyExpiryWarningDays* and *ptoken.keyRefreshWarningDays* cause Vending Key Expiry warnings on the PrismToken dashboard when these thresholds are crossed.
- Setting *account.password.maximum.age.days* determines the maximum age of a password. An account password must be changed within this period, or the account will be locked out (requiring administrator intervention).

## 8.5 Monitoring

Reliable operation requires regular monitoring. We recommend the following checks on a regular schedule (at minimum once per week):

- On the TsmWeb Licenses page check the "Expires" date under "NSS License". If the license will expire soon contact Prism to renew your license.
- On the TsmWeb Network page check the "Certificate Expires" under "SSL/TLS Server Properties". If the certificate will expire soon use the "Tasks" menu to "Generate new TLS key and certificate". The TsmWeb home page will show a warning if the certificate will expire within 30 days.
    - After generating a new certificate you will need to add the new certificate to the trusted stores of your clients, as explained in Security in a Live environment above.
- Check for alerts on the TsmWeb Alerts page.
- Check for Notices on the PrismToken dashboard. First click "Tasks" -> "Refresh" to ensure you are viewing current (not cached) information. These notices will alert you of current or pending problems with PrismToken, including:
    - The Security Module "Inhibit Vend Date" (visible in the "Security Module" section of the PrismToken dashboard) is within *ptoken.inhibitVendWarningDays*. Contact Prism to update your license.

  - The Security Module's "Transactions Left" (visible in the "Security Module" section of the PrismToken dashboard) is below *ptoken.txCounterLowThreshold*. Contact Prism to update your license.
  - One or more Vending Keys has an Issued Until (IUT) date within *ptoken.keyRefreshWarningDays*. You will need to obtain a fresh Key Load File (KLF) from the KMC.
  - One or more Vending Keys has an Expiry (EXP) or Key Expiry Number (KEN) date within *ptoken.keyExpiryWarningDays*. This could be because the Vending Key is being phased out, or the Supply Group owner no longer wants you to have the key. You will need to consider the impact of this key expiry on your vending environment, and contact the Supply Group owner(s) of the affected Vending Key(s) if these values needs to be changed.
  - One or more Vending Keys cannot vend (attribute VND=0) for a reason not related to SM License or IUT/EXP. This typically happens when the Unit Limit or Currency Limit of the Vending Key has been reached (described in more detail below). If you see a warning that keys cannot vend then you will need to inspect each key on the PrismToken Vending Keys page, identify the key(s) with VND=0, and determine the reason for each.
  - Real-time clock drift exceeds *ptoken.rtcDriftHighThreshold*. The Security Module enforces STS rules that require the token timestamp to be within a narrow window of real (wall-clock) time. If the Security Module's internal clock is very different from the clock of the NSS (or PC) this may prevent tokens from being issued. The resolution is typically to synchronise the time of the NSS (or PC) to the SM's clock (the NSS normally does this automatically). If the SM's clock is very different to wall-clock UTC time (more than 24 hours) then contact Prism for advice.
- On the PrismToken Key Management page:
  - For Vending (not Manufacturing) systems: Check that none of the KMC cards are in LOADING state. VendingKeys associated with that KMC cannot be used in vending operations when the KMC card is in LOADING state. The card should only be in LOADING state if a Key Load File has been uploaded but the loading session has not been completed (by loading VKs and ending the key agreement session).
  - Check the "expires" date of the "Public key" of each KMC. If the key is expired (or will expire soon) then you should contact the KMC for an updated Public key. You need an active (not-expired) Public key to generate a VKLOADREQ and thus to get a fresh Key Load File from the KMC.
- On the PrismToken Vending Keys page:
  - If you have any Vending Keys with a Unit Limit (ULM) or Currency Limit (CLM), then check these values to ensure that they are not close to zero. These financial risk control values limit the amount of Credit Units or Credit Currency that can be issued by the Security Module using that Vending Key. They are set to a maximum (determined by the Supply Group owner at the KMC) when the keys are refresh by loading a Key Load File, then decremented whenever a Credit token is issued. When they reach zero you will not able to issue further credit tokens. If these values are low you will need to obtain a fresh Key Load File (KLF) from the KMC.
- On the TsmWeb Users page check the "Accounts expires" and "Password expires" of every account. You can extend the account expiry date by editing the account. To extend the password expiry date you must set a new password (humans users should be instructed to change their own passwords; for API client software you will need to distribute the new password to the client).

- Check the TsmWeb Activity page. If these is lots of activity from User=$ANON, from IP addresses not on your network, or the Activity looks suspicious, then your TsmWeb/NSS may have been scanned by a vulnerability scanner. Check with your IT department whether this was an intentional security activity undertaken by them; if not then: (1) investigate whether the IP addresses indicate an internal or external scan; (2) for external scan revise your firewall rules to prevent direct access to TsmWeb/NSS; (3) for internal scan hand to your IT security team for investigation.
- On the TsmWeb home page check the “Disk Space”. If less than 100Mb is available then old data needs to be trimmed from the database. Database trimming is performed automatically by TsmWeb versions 4.70 and higher.

## 8.6 Known issues affecting reliability

The following activities are known to potentially affect reliability of PrismToken:

- TsmWeb UI operators should avoid time-consuming operators such as report generation. Such operations can block out all PrismToken API calls until completed.
- Ensure that PrismToken is not overloaded by API calls. If you see a consistent increase in call latency this could indicate that you are at PrismToken’s maximum performance.
- We have a few reports of an intermittent problem when loading a Key Load File that can put the SM into LOADING state but without Vending Keys available to load. This is potentially serious in a Live environment, and may require fail-over to a standby/DR system. We are investigating the problem but have been unable to reproduce it.
- The NSS is an embedded system with limited resources, and various internal protections against malicious modification that can progressively consume these resources when the NSS is under load for extended periods (as would be expected in a Live environment), potentially resulting in a system freeze. To ensure reliable operation we suggest a scheduled reboot of the NSS approximately once per week at an off-peak time. We are investigating resolutions for this problem.

# 9 Amendment History

| Version | Description | Person | Date |
|---|---|---|---|
| Draft A | Created first draft | SG | 2020/06/05 |
| Draft B | Second draft | SG | 2020/09/15 |
| Final Draft | Final draft edited | SG | |
| Revision | Revision based on feedback | GG | 2020/09/19 |
| 1.0 | Formatted into correct template | TD | 2020/09/21 |
| 1.1 | Added "Operations and Monitoring" section | TD | 2021/09/06 |
| 1.2 | Added "Load All VKs" section under vending keys and added enabling PrismToken in a meter manufacturing environment | SG | 2021/09/07 |
| 1.3 | Updates and corrections to section 6 regarding the DITK | SS | 2022/02/01 |
| 1.4 | Added Section 5.6.3 which explains how to load individual VKs | CA | 2022/02/15 |
| 1.5 | Collection Manufacturing setup instructions into section 4.1.1 | TD | 2022/05/19 |