APPLICATION NOTE: AN-1240

AC500 V3 - ENCRYPTION AND CERTIFICATES
INTRODUCTION AND CONFIGURATION
Contents

1 Introduction ................................................................................................................................. 3
  1.1 Scope of the document ............................................................................................................ 3
  1.2 Compatibility .......................................................................................................................... 3
  1.3 Abbreviations ........................................................................................................................ 3

2 Basics about encrypted communication .................................................................................. 4
  2.1 Symmetric & asymmetric encryption ..................................................................................... 5
  2.2 TLS Handshake ...................................................................................................................... 6
  2.3 Signing Certificates for trusted communication ................................................................. 7
  2.4 References ............................................................................................................................... 8

3 Security Screen in Automation Builder ...................................................................................... 9
  3.1 User ........................................................................................................................................ 9
  3.2 Project .................................................................................................................................... 9
  3.3 Device ................................................................................................................................... 11
     3.3.1 Create self-signed certificates ....................................................................................... 12
     3.3.2 Create an CA-signed certificate .................................................................................... 12
     3.3.3 Add Trusted root certificates ....................................................................................... 14

4 Encrypted communication & webserver .................................................................................. 16
  4.1 Change PLC Runtime Policy ............................................................................................... 16
  4.2 Enable encrypted communication ........................................................................................ 16
  4.3 Encrypted Webserver access ............................................................................................... 17
     4.3.1 Certificates signed by a public CA ............................................................................... 18
     4.3.2 Certificates signed by an internal CA ......................................................................... 18
     4.3.3 Self-signed Certificates ............................................................................................. 18
  4.4 Encrypted webserver to CP600 with Chromium browser .................................................... 19
  4.5 Encrypted communication to CP600 with PLC Handler ..................................................... 21

5 OPC UA ..................................................................................................................................... 22
  5.1 Create Server Certificates .................................................................................................... 22
  5.2 Setting up an encrypted connection with the ‘UaExpert’ Client ......................................... 22
  5.3 Encrypted communication to UA Expert ............................................................................ 26
  5.4 Further development ............................................................................................................ 26

6 FTPS ........................................................................................................................................ 27

7 Additional Examples ................................................................................................................. 28
1 Introduction

1.1 Scope of the document
This document explains some background information about encrypted communication. It shows how encryption can be configured using Automation Builder and AC500 V3 PLC or third-party software.

Further details regarding cyber security can be found in our whitepaper and the cyber security FAQs.

In addition to encrypted communication, user management can be configured in the PLC. User Management allows only authorized users to log in, see the webvisu, access the OPC UA data and so on. For further details please check the application note User Management.

1.2 Compatibility
The application note explained in this document has been used with the engineering system versions below. They should also work with other versions, nevertheless some small adaptations may be necessary for future versions.

- AC500 V3 PLC
- Automation Builder 2.6.1 or newer

1.3 Abbreviations
CA Certificate Authority
CSR Certificate Signing Request
FTPS File Transport Protocol Secure
OPC UA Open Platform Communication Unified Architecture
PKI Private Key Infrastructure
TCP Transmission Control Protocol
TLS Transport Layer Security
2 Basics about encrypted communication

This chapter explains the basics about encryption and how a secure communication between two components is established using an TLS (Transport Layer Security) handshake.

Whenever the communication must be private between two components encrypted communication is required. This is especially required when using an authentication with username and password or a process that can be monitored or controlled via the internet.

If encryption is not used attackers can easily get passwords or take over control of the process.

As an example, two people, Bob and Alice, want to communicate with each other. A third person, Eve, is an attacker who tries to get access to this communication. This means they want to achieve the following security goals:

1. Bob and Alice want to be sure that no third person like Eve can read the messages. Also known as confidentiality.
2. Bob and Alice want to be sure that no third person like Eve can manipulate the messages. Also known as integrity.
3. Bob wants to be sure he is really talking to Alice and Alice herself wants to ensure the message is really from Bob. Also known as authenticity.
4. Both Bob and Alice can be sure that the messages they are sending are also received by the other person. And the other person cannot deny that they received it. Also known as liability.
2.1 Symmetric & asymmetric encryption

There are two ways to encrypt messages.

Using a symmetric encryption means both sides have exactly the same key. This key is used for encryption and decryption of the message.

In contrast to symmetric encryption a keypair is used for asymmetric encryption.

The encryption key is used to encrypt the message. But there is no way to decrypt the message with this key. Therefore, the decryption key must be used. These are also known as public and private keys. The encryption key is public so everybody can encrypt a message. But only the person with the decryption key can read the message again.
2.2 TLS Handshake

The TLS handshake is based on a Diffie-Hellmann key exchange. The basics of this are explained in this chapter in a simplified way. Further information about TLS and Diffie-Hellmann can be found in chapter 2.4 References.

If Bob and Alice both have the same symmetric key, they exchange messages with each other and nobody except the two of them can decrypt the message. So a confidential and integral communication between Bob and Alice is possible.

The problem is: How can Bob and Alice exchange this symmetric key so that no one else, like Eve, is able to access the key. If Bob would just send the Symmetric key to Alice, Eve could also get it. To exchange the symmetric key, asymmetric keys are used.

Alice owns an asymmetric keypair. The encryption key is public and can be shared with everybody while the decryption key is private and belongs to Alice. As visible in the picture above Bob first sends a “Hi” to Alice to start the conversation. She has a keypair and sends the public encryption key to Bob.

Bob can encrypt a message with Alice public key and send it to her. As only Alice owns the private key, she is the only person who can read the message. The same way Alice could encrypt a message with Bobs public key. The problem is that the encryption and decryption with asymmetric keys is complex and time consuming. Therefore, the goal is to use a symmetric key for encryption.

After receiving Alice public key Bob is not encrypting the message, he wants to send but generates a symmetric key and encrypts this symmetric key with the public key from Alice.
Alice receives this message and decrypts it to get the symmetric key. Now both have the symmetric key and can start a symmetric encrypted communication. In the real system the Diffie-Hellman key exchange is used which is different and more complex than the key exchange explained here.

If Eve was listening, she could read the Hi and the public key from Alice in clear text but the symmetric key from Bob she cannot see. Therefore, Eve is not able to follow the encrypted communication as she is missing the symmetric key. The communication is confidential and integer.

But if Eve redirects Bob's message to herself, Bob talks to Eve even he thinks he's talking to Alice. Then he might share confidential information with Eve.

To ensure that Bob is really talking to Alice, he needs to verify Alice authenticity. Therefore, Alice requires a signed public key to prove that she is really Alice. The signature process is explained in the chapter below.

### 2.3 Signing Certificates for trusted communication

A certificate is something like a personal ID card. The ID card is provided by the government and only valid for a specified period.

A certificate authority (CA) is an entity that signs digital certificates. The most common CAs are IdenTrust, DigiCert and Sectigo. A certificate used for encryption, like required in this example must be signed by a CA.

Here Alice had generated asymmetric keys. The public key she sends together with some identifying information to the CA. This is called Certificate Signing Request (CSR).
The CA owns their own key pair. Alice CSR will be validated by the CA and signed with the private key of the CA to declare it as trustful. Once the signature is placed by a CA the information in the certificate cannot be changed. This signature is only valid for a specified period. After the period ends the certificate will expire and will not be trusted anymore. Therefore, it’s required to renew the certificate before it expires.

Because the signature is created with the private key of the CA nobody else can sign it. But the CA’s public key everybody can use to check that the signature is valid.

After Alice’s public key was signed it is sent back to Alice. Because the CA signed Alice’s public key they are certifying Alice authenticity. Now the public key is CA-signed instead of self-signed.

Certificates can be requested for different purposes, e.g.:

- Digital signatures
- Code signing

During the TLS handshake Alice now sends the signed certificate to Bob. He is now able to check with the public key from the CA whether Alice is trustworthy or not. Depending on this he continues the communication or not.

Compared to a normal industrial use case Alice is a server in the network and Bob a client who is connecting. Alice has the server certificates that authorize that the server is trustworthy. Bob as the client can check the server’s authenticity and continue with the connection. If Bob as the client needs to authorize himself to be able to communicate with Alice as the server, he also needs a certificate. These certificates are called client certificates. If both Bob and Alice – client and server – are using signed certificates the mutual authenticity of the communication is given.

2.4 References

Further details about Encryption, TLS and CAs can be found in the literature. The literature used for this Application Note is listed below.

- Michael Driscoll - The Illustrated TLS Connection: https://tls12.xargs.org/ Accessed 22.11.2023
3 Security Screen in Automation Builder

The security screen is handling all settings related to security. It can be found via View > Security Screen or in the bottom right corner of Automation Builder.

The security screen is split into three subtopics User, Project and Device which are explained in the next chapters.

A detailed description is available in the online help.

3.1 User

User certificates are part of one computer/ Automation Builder installation. These user certificates can be used to encrypt and sign the following items:

- Project
- Library
- Boot application – see Encrypt and Sign the boot application.

Furthermore settings to enforce encryption and communication settings for this user can be changed.

It is possible to handle different profiles here. For each profile, certificates for code signing and project decryption can be added. Encryption and signing can be enforced for communication with the PLC, the project and the boot application.

Detailed information can be found in the online help.

3.2 Project

In the project subtopic there are settings which are part of this Automation Builder project. As an example this can be the encryption of the project with a password which is explained below. The rest of the settings are explained in the online help.

Encrypt Project with password

1. In Automation Builder go to the Project menu and choose Project Settings
2. Choose Security

3. Select Encryption and Password and type in your Password

CAUTION!
If you forget the Password, you will have no way to access the project.
3.3 **Device**

In Devices the certificates of the AC500 V3 PLC are managed.

Own certificates are client or server certificates of the PLC where both public and private key are available inside the AC500 PLC.

Trusted certificates are the public certificates from the CA’s which should be trusted.

If certificates should be revoked, they can be moved to Untrusted or Quarantined.
### 3.3.1 Create self-signed certificates

1. Set the Gateway to connect to the PLC
2. Set the correct date and time in the PLC

| Note: For AC500 a battery is required to store the time after power cycle. An NTP client can be used to set the time automatically |

3. Open the Devices Tab in the security screen
4. Click the PLC name here “PLC_AC500_V3”
   A list of possible Certificates for this PLC is listed
5. Select the Certificate you want to create, which is not yet available. Here “Web Server”
6. Click “Create a new certificate on the device”
7. Key length must be 4096 bit
8. Validity period is by default 1 year and can be changed between 1 day and 10 years
9. Click OK to create an asymmetric key pair which is self-signed

<table>
<thead>
<tr>
<th>CAUTION!</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificates have a validation period. In case the PLC time is not set the certificate might expire. To ensure the PLC is up to date an NTP client can be used. If no NTP server is reachable from the network the time must be buffered with a battery in the standard PLC.</td>
</tr>
</tbody>
</table>

Further details can be found in the online help.

### 3.3.2 Create an CA-signed certificate

1. Create self-signed certificates as mentioned above.
2. Double click the PLC main node and select PLC Shell from the tabs
3. Type `cert-getapplist`
Here the index of the WebServer is 4

4. Type `cert-createcsr <index>` here `cert-createcsr 4`

```
cert-createcsr 4
Create ASN.1 CSR for application with index 4. Check logger to see when finished.
```

5. Change to the tab “Log” and refresh the view

The entry "[<index>] CSR file created, subject =‘commonName=<PLC Name>’" should be available.

6. Change to Files tab and browse on the PLC to cert/export

7. Copy the *.csr file somewhere to your PC

8. The request needs to be signed by a certificate authority (CA), and then you receive a signed certificate from the CA.

9. In the security screen Select Devices > Own certificates and click download to device

More details can be found in the online help.
3.3.3 Add Trusted root certificates

If the PLC as a client is connecting to a server via encrypted communication, the PLC needs to verify the trust of the server. For servers which are trusted the root CA certificate should be uploaded to Trusted certificates.

This chapter explains how to import the DigiCert Global Root C2 Certificate (required for Azure IoT hub) to the cert store in the PLC

1. Open certmgr on your pc

2. Select the required certificate here "DigiCert Global Root G2" in Trusted Root Certification Authorities > Certificates

3. Click Action > All Tasks > Export

4. Click Next and select "Base-64 encoded X.509 (.CER)"
5. Click Next and select a folder where to store the certificate
6. Click Next and Finish the Export
7. Go Back to the security screen in Automation Builder
8. In the Tab Devices select the folder “Trusted Certificates”

9. Click the Download button and select the certificate which shall be imported
4 Encrypted communication & webserver

A signed certificate is required for encrypted communication and the webserver must be available as described in chapter 3.3.1 and 3.3.2.

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

4.1 Change PLC Runtime Policy

By default, the PLC is not configured for encrypted communication.

1. Open the communication settings and establish the connection

2. Click Device > Change Runtime Security Policy …

3. Change Communication policy from No encryption to Optional or Enforced encryption

4. Confirm with OK

4.2 Enable encrypted communication

1. Logout from the PLC

2. Click Device > Encrypted Communication
3. Colored Line is showing that Encryption will be used

4. Establish the connection again

5. A Warning is displayed when using self-signed certificates that are not signed by a public CA.

   ![Automation Builder 2.6 - Premium](image)

   It is possible to install the certificate to Controller Certificates on your PC. With the certificate installed the warning will not appear again for this PLC or set it up to accept the certificate only for this session.

6. Push OK and Login

   PC and PLC are now connected via encrypted communication

4.3 **Encrypted Webserver access**

By default, the webserver supports HTTP and HTTPS. The security mode in the webserver can be changed in the webserver settings. When selecting: Redirect HTTP to HTTPS requests for non-encrypted communication will be also forwarded to Port 443 and an encrypted communication will be set up.

Note: The webserver settings are boot parameters. A change of these settings requires an additional PLC reboot after downloading.
4.3.1 Certificates signed by a public CA
When connecting to the webserver via any browser and using CA signed certificates the closed lock in Chrome shows that the communication is encrypted.

4.3.2 Certificates signed by an internal CA
When certificates are used which are not publicly signed but signed from an internal CA the CA signing and root certificate needs to be added to the trusted certificates on your PC.
To do this open “Manage user certificates” on your PC. Browse to the folder Trusted Root Certification Authorities > Certificates and click Action > Import

Follow the wizard to import the CA certificates.

4.3.3 Self-signed Certificates
When using self-signed certificates an error will be displayed, that the webserver is not trusted. If you still want to continue follow the steps below

Google Chrome, Edge or any other chromium based browser

Click somewhere and type “thisisunsafe” and hit Enter
Encrypted connection to webserver will be done but showing Not secure
4.4 Encrypted webserver to CP600 with Chromium browser

How the Chromium browser is activated is described in this Application note.

The browser will automatically use encrypted communication if the PLC is prepared for it.

All settings and views are the same as described in the last chapter for the Chromium Browser.

In the case of using Certificates signed by an internal CA these certificates must be added to the cert store in the panel.

1. Store the certificates on an USB device
2. Open the CP600 System Settings and log in as administrator
3. Click Open certificate manager
4. Switch to the tab Authorities and click Import

![Image of Privacy and security settings with Authorities tab highlighted]

5. Click Home on the left-hand side and navigate to mnt > usbmemory

![Image of file system with mnt/usbmemory directory highlighted]

6. Select the certificate from the usb device and install it as trusted for identifying websites

![Image of certificate authority settings with trust options selected]

The certificate "AC500 HD PLC Test Signing CA" represents a Certification Authority.
### 4.5 Encrypted communication to CP600 with PLC Handler

Open the security screen in Automation Builder

Select the certificate for Encrypted communication and click the upload from device to PC button

Save the certificate on the hard disk of your PC

Open the Panel Builder project and use the CODESYS V3 Handler as protocol

Set the IP address and if required a username and a password for the PLC user management

Click the … next to Certificate and select the certificate which was exported in step 0

The certificate will be inserted as text in the field

Add tags and use them as known for the CODESYS V3 Handler via a Symbol file. Communication via V3 Handler will be encrypted from now on.
5 OPC UA

5.1 Create Server Certificates

To encrypt data and exchange it with the client (e.g. Unified Automation UaExpert) safely, the server needs a certificate that the client must classify as trusted when a connection is established for the first time.

Requirement: The active path to the controller is set. And the Device Certificate for the OPC UA Server exists. See chapter 3.3.1 and 3.3.2

Select the OPC UA Server certificate and upload the Certificate to your PC

5.2 Setting up an encrypted connection with the ‘UaExpert’ Client

The UaExpert OPC UA Client is freely accessible software that you can download from the internet. You can use this client to connect to the V3 OPC UA Server. The following description refers to this program. Other OPC UA Clients work in a similar way.

1. Start the Client UaExpert program. On the first start, the configuration tool will create a UA application instance certificate. Therefore, a lot of information is requested and must be set in the dialog.
After all sections are filled out, the application instance certificate can be generated by pressing the ‘OK’ button.
2. To add and connect a UA Server to the UaExpert. Just click on Add Server to add a connection to a new UA Server. The ‘Add Server’ dialog opens:

3. In the tree view, choose the field under Custom Discovery and input the URL from the connected OPC UA server.

4. After that, expand the option OPCUAServer@PM5650-2ETH
5. Select the connection type Basic256Sha256-Sign & Encrypt(uatcp-uasc-usbinary) and click OK to close the dialog.
6. Click option ‘Connect’ from the menu Server, The Certificate Validation dialog opens with an error message. Select the ‘Accept the server certificate temporarily for this session’ option and push the Continue button.

7. Switch back to the devices of the Automation Builder security screen and go to the Quarantined Certificates folder.

8. Drag the certificate (e.g. UaExpert) to the Trusted Certificates folder.

Note: Here a self-signed certificate from UA Expert is used, therefore the certificate must be shifted from Quarantined to Trusted. In a real system the certificate should be CA-signed. Then the root certificate from the OPC UA server can be directly installed as a Trusted Certificate.

5.3 Encrypted communication to UA Expert

In the UaExpert client, Click the option ‘Connect Server ‘again. Do the same with the above sub-chapter 5.2.6.

The connection is established, and objects are displayed in the data success view. Communication via OPC UA will be encrypted from now on.

5.4 Further development

- Connecting to an AC500 V3 OPC UA Server with client certificates. Planned for AB 2.8
- Encrypted communication with AC500 V3 PLC as OPC UA client. Planned for AB 2.8
6 FTPS

To enable an encrypted FTPS connection a certificate for the FTP server must be in the PLC. How to do this is explained in chapter 3.3.

After adding an FTP server below the ETH interface, the security mode can be changed to Both or FTPS only.

After downloading an FTPS client like FileZilla, an encrypted communication to the PLC can be established.
7 Additional Examples

Further Application Examples and Notes are available in the ABB library.

- White paper
- Cyber Security FAQs
- User Management
- Certificate Store access from IEC code
We reserve the right to make technical changes or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB AG does not accept any responsibility whatsoever for potential errors or possible lack of information in this document.

We reserve all rights in this document and in the subject matter and illustrations contained therein. Any reproduction, disclosure to third parties or utilization of its contents – in whole or in parts – is forbidden without prior written consent of ABB AG.

Copyright© 2024 ABB. All rights reserved.