Tampering Protection

# Link

<https://www.youtube.com/watch?v=qCuVBD2dmTA&list=PLnMKNibPkDnFzux3PHKUEi14ftDn9Cbm7&index=27>

# Description

In this paperwork, we will learn how to use asymmetric encryption on our board.

Contents

[Link 1](#_Toc90314842)

[Description 1](#_Toc90314843)

[Prerequisites 2](#_Toc90314844)

[STM32 Board 2](#_Toc90314845)

[ST-Link cable 2](#_Toc90314846)

[STM32CubeMX 2](#_Toc90314847)

[STM32CubeIDE 2](#_Toc90314848)

[Walkthrough 3](#_Toc90314849)

[Step 1 : Launch STM32CubeMX and generate the code 3](#_Toc90314850)

[Step 2 : Write the main 4](#_Toc90314851)

[Step 3 : Check the encryption 5](#_Toc90314852)

# Prerequisites

A picture containing chart

Description automatically generated

## STM32 Board

## ST-Link cable

## STM32CubeMX

## STM32CubeIDE

# Walkthrough

## Step 1 : Launch STM32CubeMX and generate the code

Launch STM32CubeMX and select the right board depending on the one you are using. In my case I use the WB55 Nucleo board. Then you can generate the code of your project.  
Then you have to activate PKA.  
  
Graphical user interface, application

Description automatically generated

Step 2 : Generate the private and public key.  
To generate your keys, you have to install openssl first. Once it is installed, you can generate the following commands using the application to generate them.  
Generate private key :   
A screenshot of a computer

Description automatically generated with medium confidence  
Read private key components : Text

Description automatically generated  
Generate public key :  
Step 3 : Write the main.c

When you have the informations about your private key, just put them in the same format than in the screenshot. You have to enter the modulus, the public exponent, the private one, and a plaintext to encrypt.  
You can use apps like notepad to format the values in the correct format.  
We will also set a variable for the encryted and decrypted data.  
Text

Description automatically generated with low confidence  
Then add a variable in the main function to set your PKA parameters.  
Text

Description automatically generated with medium confidence  
Then we will set the code for the encryption and the decryption. We basically set the values and just run the encryption. We set an error handler if something unexpected happens.  
Graphical user interface, text, application

Description automatically generated

Step 4 : Check the encryption

Finally we just have to put our variables in the debug expressions and check them after the program has been launched.  
Table

Description automatically generated  
You could normally see that the plaintext is equal to the decrypted data.