

OPTIGA™ TPM Application Note

Integration of TLS Functionality for OPTIGA[™] TPM SLx 9670 TPM 2.0

Devices

- OPTIGA[™] TPM SLB 9670 TPM2.0
- OPTIGA[™] TPM SLI 9670 TPM2.0
- OPTIGA[™] TPM SLM 9670 TPM2.0





About this document

Scope and purpose

The world we live in is a connected world. Today we rely on our phones, computers and connected devices to communicate, buy goods, travel and work. It is expected that these devices increase exponentially.

All these devices that are connected to the internet have one thing in common – They rely on the protocol called TLS (Transport Layer Security) to protect their information in transit.

TLS is a cryptographic protocol designed to provide secure communication over an insecure infrastructure. This means that, if this protocol is properly deployed, you can open a communication channel to an arbitrary service on the internet and be reasonably sure that you're talking to the correct server, and exchange information safely knowing that your data will not fall into the wrong hands and that it will be received intact.

This is not the case in the real world. Complex systems along with software bugs can open a back door to an attacker. Aside from this, the simplicity of the RSA algorithm (which is widely used in most of the systems running TLS), has known weaknesses, such as the Private Keys being stored in software.

Anyone with the access to the corresponding private key can decrypt the communication between the client and it. This type of attack does not need to happen in real time. An attacker could establish a long-term operation and record all the encrypted traffic and wait until he obtains the Key. After the Key has been compromised, it's possible to decrypt all previously recorded traffic.

The OPTIGA[™] TPM SLx 9670 TPM2.0 is a hardware security controller fully compliant with TCG TPM products with CC (EAL4+) and FIPS certification that can be used to harden the TLS connection by managing and keeping the Private Keys secure amongst other security features.

The OPTIGA[™] TPM SLx 9670 TPM2.0 products, standard, automotive and industrial versions, differ with regards to supported temperature range, lifetime, quality grades, test environment, qualification and reliability to fit the target applications requirements. For more details refer to Infineon's website [1].

This document contains an overview of what is Transport Security Layer (TLS) and step-by-step instructions on how to use the TLS Stack Software with the Trusted Platform Module OPTIGA[™] TPM SLx 9670 TPM2.0 on a Raspberry Pi[®] 3B+/4 Linux environment by using a Server-Client connection.

The described steps to integrate an OPTIGA[™] TPM in a Raspberry Pi[®] 3B+/4 Linux environment can be performed with one of the Infineon Iridium SLx 9670 TPM2.0 SPI Boards, listed in the Table below.

Iridium Boards:

Supported TPM	Order type	Order number
OPTIGA [™] TPM SLB 9670 TPM2.0	IRIDIUM 9670 TPM2.0	SP001596592
OPTIGA™ TPM SLI 9670 TPM2.0	IRIDIUM SLI 9670 TPM2.0	SP004232000
OPTIGA™ TPM SLM 9670 TPM2.0	IRIDIUM SLM 9670 TPM2.0	SP004232004

The 3 Infineon Iridium Boards are referred in the following as "Infineon Iridium SLx 9670 TPM2.0 SPI Board"



Intended audience

This document is intended for customers who want to increase the security level of their embedded platforms using a OPTIGA[™] TPM SLx 9670 TPM2.0 from Infineon Technologies in combination with the Open Source TPM Software Stack 2.0 (TSS 2.0) and like to evaluate how to incorporate TLS with the TPM for their target applications.

This application note was tested using a Raspberry Pi[®] 3B+/4 and Raspbian Buster with desktop and recommended software with kernel version 4.19 [7].



Table of contents

Abou	It this document	
Table	e of contents	
List o	of figures	5
1	The Transport Layer Security (TLS 1.2)	6
2	TLS Hardening by OPTIGA™ SLx 9670 TPM2.0	7
2.1	Using SLx 9670 TPM2.0 to Harden the TLS Session	7
2.2	tpm2-tss-engine OpenSSL Plug-In	7
2.3	Using tpm2-tss-engine	8
2.3.1	Sanity-Test	8
2.4	OpenSSL Version	9
2.5	Using OpenSSL and the TPM2-TSS Engine to Create PKI Used in TLS session	9
2.5.1	TPM 2.0 Key Management	9
2.5.2	Creating OpenSSL Configuration File	10
2.5.3	Creating the Root CA and Its Certificate	13
2.5.4	Creating the Intermediate CA and Its Certificate	16
2.5.5	Creating Client/Endpoint Key Pair Using SLx 9670 TPM2.0	19
2.5.6	Creating Client/Endpoint CSR Using SLx 9670 TPM2.0	19
2.5.7	Signing Client/Endpoint CSR with RootCA	21
2.5.8	Creating Server Certificate	22
2.6	Creating an OpenSSL S_Server	23
2.7	Creating an OpenSSL S_Client	24
3	Decoding SSL/TLS Traffic using TShark	29
3.1	Installing TShark	29
3.2	Available Network Interfaces to use with TShark	29
3.3	Testing the Capture of Network Traffic with TShark	30
3.4	Capturing a TLS Session using TShark	31
4	References	
Revis	sion history	40



List of figures

List of figures

Figure 1	Sanity test for TSS Engine	9
Figure 2	OpenSSL version	9
Figure 3	TPM 2.0 Key Wrapping	9
Figure 4	Creating directory structure	10
Figure 5	OpenSSL Configuration File	10
Figure 6	Copying openssl.cnf reference file	13
Figure 7	Root CA Distinguished Name or DN	14
Figure 8	Root CA Certificate	15
Figure 9	Creating Intermediate CA CSR	16
Figure 10	Intermediate CA Certificate Generation	17
Figure 11	Verify Intermediate CA vs. Root CA	17
Figure 12	Intermediate CA Certificate	18
Figure 13	Client/Endpoint Key Pair wrapped by TPM	19
Figure 14	Client/Endpoint CSR Information	20
Figure 15	Client/Endpoint CSR	21
Figure 16	Signing Client CSR with Intermediate CA	21
Figure 17	Client Certificate Chain verification	21
Figure 18	Client/Endpoint Certificate	22
Figure 19	Server Certificate	23
Figure 20	Client Certificate Chain verification	23
Figure 21	OpenSSL S_Server	24
Figure 22	OpenSSL S_Client and S_Server TLS Handshake hardened by OPTIGA [™] SLx 9670 TPM2.0	25
Figure 23	OpenSSL S_Client and S_Server TLS Cipher	26
Figure 24	OpenSSL S_Client and S_Server TLS session flow	27
Figure 25	TShark Install	29
Figure 26	Network Interface detected in Raspberry Pi [®] 3B+/4	29
Figure 27	TShark capturing terminal window	30
Figure 28	Terminal window 2	30
Figure 29	Opening S_Server	31
Figure 30	Start of TShark capture task	31
Figure 31	S_Client transaction	32
Figure 32	TShark captured packets	32
Figure 33	Reading tls.pcap file using TShark	33
Figure 34	TLS Client Hello	34
Figure 35	TLS Server Hello	35
Figure 36	TLS Handshake Protocol Certificate	36
Figure 37	TLS Creation of Session Ticket : All communication are encrypted at this point	37



1 The Transport Layer Security (TLS 1.2)

TLS are cryptographic protocols designed to provide secure communication over insecure infrastructure. When these protocols are properly deployed, you can open a communication channel to an arbitrary service on the Internet and have certain level of security and assurance that it will be talking with the correct server and exchange information safely (confidentiality). These protocols protect the communication link.

These are the main objectives of TLS:

- Cryptographic Security
 - Enables Authentication, Confidentiality and Integrity in a communication between two parties that exchange information.
- Interoperability
 - TLS protocols are not system dependent. They can be used for example in Linux, Android, Bare Metal systems.
- Extensibility
 - TLS is a framework for the development of cryptographic protocols. TLS looks to be independent from cryptographic primitives, like ciphers and hashing functions.

We will discuss in section 2.1 [Using SLx 9670 TPM2.0 to Harden the TLS Session] some of the disadvantages of using software based TLS libraries solely and how to achieve a higher level of security and assurance when using SLx 9670 TPM2.0.



TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0 2

At the heart of every TPM software implementation there is the TPM Software Stack (TSS). It is a middleware to support, improve and simplify TPM usage for programmers. In general, the TSS features a layered design to fit various requirements from application developers by providing different user APIs with varying abstraction layers and functionality.

2.1 Using SLx 9670 TPM2.0 to Harden the TLS Session

TLS is used by web services and IoT devices to transmit sensitive information between client/Endpoint and Server/Cloud applications. TLS provides authenticated key exchange using asymmetric cryptography, data confidentiality using symmetric encryption and message integrity using message authentication codes scheme. However, these crypto primitives are stored in system memory and do not provide any trustworthiness assurance of the involved endpoint.

The drawback is that their implementation is using software library modules that store private keys in application or secure memory and have proven to contain bugs or vulnerabilities which have been exploited for the last several years.

By using SLx 9670 we can embed crypto operations inside dedicated fixed TPM 2.0 function calls used by TLS protocols, like for example using TPM 2.0 as the source of entropy for the TLS required random number.

The scope of this application note is to show the benefit of using SLx 9670 TPM2.0 to protect the private key involved in the TLS handshake process. This is only one of different ways SLx 9670 TPM2.0 can help harden a TLS session. Token Biding is another.

Token Binding is an extension to TLS that provides stronger authentication and longer, more robust sessions.

The idea behind token binding is "proof of possession". The challenge with tokens is that they're only as secure as where you store them. Token binding uses cryptographic key pairs and TPM 2.0 for secure storage.

With token binding, man in the middle attacks can't forward requests or replay credentials because they can't prove they have the key bound to the token.

Token Binding will not be covered in this application note. For more information about Token Binding, refer to [9]

tpm2-tss-engine OpenSSL Plug-In 2.2

Enhanced System API (ESAPI) is part of the open source software stack for TPM 2.0 "TSS". It makes work easier for developers who want to use the Trusted Platform Module (TPM) 2.0 - a standardized hardware-based security solution for securing industrial, automotive and other applications such as network equipment. This is the first open source TPM middleware that complies with the Software Stack (TSS) Enhanced System API (ESAPI) specification of the Trusted Computing Group (TCG), providing significant value to the open source community.



TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

The ease of integration on Linux and other embedded platforms comes with the release of the TPM 2.0 ESAPI stack which speeds up the adoption of TPM 2.0 in embedded systems such as network equipment and industrial systems.

You can find the ESAPI TPM 2.0 TSS stack within the **tpm2-software** open-source project [2].

Note: Before moving on with this application note a pre-requisite is to have the following components of the **tss2-software** package installed:

- TPM Software Stack 2.0 (tpm2-tss)
- o TPM2 Access Broker & Resource Manager (tpm2-abrmd)
- TPM2 Tools (tpm2-tools)
- TPM2 TSS Engine (tpm2-tss-engine)
- Cryptsetup (cryptsetup)

The details on how to install and test these packages are part of the documentation on GitHub [2] and can be found within the TPM Evaluation Kit. For more information about the TPM Evaluation Kit, please get in touch with your local sales.

The **tpm2-tss-engine** project implements a cryptographic engine for **OpenSSL** [3] for **Trusted Platform Module (TPM 2.0)** [1] using the **tpm2-tss** [2] software stack that follows the Trusted Computing Groups (TCG) **TPM Software Stack (TSS 2.0)** [5]. It uses the **Enhanced System API (ESAPI)** [6] interface of the TSS 2.0 for downwards communication. It supports RSA decryption and signatures as well as ECDSA signatures.

2.3 Using tpm2-tss-engine

We will be using the tpm2-tss-engine as an OpenSSL engine to harden the TLS channel using TPM 2.0.

The development platform used for this task will be the Raspberry Pi[®] 3B+/4 along the OPTIGA[™] IRIDIUM 9670 TPM2.0 board.

To demonstrate the hardening of the TLS session between a Client/Endpoint and Server/Cloud the OpenSSL S_Server and S_Client modules will be used along with the local host capability of Linux running on Raspberry Pi® 3B+/4.

2.3.1 Sanity-Test

The following command can be executed to check if the tpm2-tss-engine has been installed successfully.

Code Listing 1 Sanity test for the TSS Engine

001 openssl engine -t -c tpm2tss

It should retrieve the engine information about the name and the available functions as shown in Figure 1.

OPTIGA™ TPM Application Note

Integration of TLS Functionality for OPTIGA[™] TPM SLx 9670 TPM 2.0



TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

```
pi@raspberrypi:~ $ openssl engine -t -c tpm2tss
(tpm2tss) TPM2-TSS engine for OpenSSL
[RSA, RAND]
[ available ]
```

Figure 1 Sanity test for TSS Engine

2.4 **OpenSSL Version**

Usage of S_Server with HSM-protected private keys is only supported on OpenSSL 1.1.0 and newer. To check the installed version of OpenSSL, run the following command.

pi@raspberrypi-os:~ \$ openssl version OpenSSL 1.1.0k 28 May 2019

Figure 2OpenSSL version

2.5 Using OpenSSL and the TPM2-TSS Engine to Create PKI Used in TLS session.

2.5.1 TPM 2.0 Key Management

The Trusted Platform Module (TPM) greatest strength is to enable an application the use of cryptographic keys while keeping them safe inside the TPM. It can both generate and import externally generated keys.

Each key has individual security controls, which can include a password or an enhanced authorization policy. These keys can be certified by the TPM and used to certify other keys as well.

In order to manage internal memory efficiently, the TPM, has the capability to wrap keys (encrypt) with the parent Key and store them (the encrypted key) outside TPM and still not compromising the overall security of the system. When the time comes to use this key, the wrapped Key is loaded into the TPM. Only the specific TPM used to wrap the key can unwrap it and use it, as shown in Figure 3.







2.5.2 Creating OpenSSL Configuration File

To establish a TLS session between a Client/Endpoint and the Server/Cloud with OpenSSL and TPM2-TSS engine we need to create an OpenSSL configuration file. This file will have the configuration we will be using to create our rootCA, Server/Cloud, Client/Endpoint Certificates and CSRs. These are part of a Public Key Infrastructure (PKI).

Create a working directory called "**tpm_hardened_tls**". Within this directory create the "**pki**" directory structure and the "**tpm2**" directory structure.

Code Listing 2

001	mkdir tpm hardened tls
002	cd tpm hardened tls
003	<pre>mkdir -p pki/{csr,certs,crl,newcerts,private}</pre>
004	<pre>mkdir -p tpm2/{csr,certs,tpm_wrapped_keys}</pre>

```
pi@raspberrypi-os:~ $ mkdir tpm_hardened_tls
pi@raspberrypi-os:~ $ cd tpm_hardened_tls/
pi@raspberrypi-os:~/tpm_hardened_tls $ mkdir -p pki/{csr,certs,crl,newcerts,private}
pi@raspberrypi-os:~/tpm_hardened_tls $ mkdir -p tpm2/{csr,certs,tpm_wrapped_keys}
pi@raspberrypi-os:~/tpm_hardened_tls $
```

Figure 4 Creating directory structure

Create a new file with the name "**openssl.cnf**", "**index.txt**", "**index.txt.attry**", under the tpm_hardened_tls directory.

Code Listing 3

001	touch ./pki/openssl.cnf	
002	touch ./pki/index.txt	
003	touch ./pki/index.txt.attry	

pi@raspberrypi-os:~/tpm_hardened_tls \$ touch ./pki/openssl.cnf
pi@raspberrypi-os:~/tpm_hardened_tls \$ touch ./pki/index.txt
pi@raspberrypi-os:~/tpm_hardened_tls \$ touch ./pki/index.txt.attry

Figure 5 **OpenSSL Configuration File**

Then copy and paste the OpenSSL Configuration File [Code Listing 4] into "openssl.cnf".

Code Listing 4 OpenSSL Configuration File

```
# OpenSSL intermediate CA configuration file.
# Copy to `/root/ca/intermediate/openssl.cnf`.
[ ca ]
```



TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

```
`man ca
default ca = CA default
[ CA default ]
# Directory and file locations.
dir
                = ./pki
                = $dir/certs
certs
crl dir
               = $dir/crl
new_certs_dir = $dir/newcerts
database
               = $dir/index.txt
                = $dir/serial
serial
RANDFILE
                = $dir/private/.rand
# The root key and root certificate.
private key = $dir/private/rootCA.key
certificate
               = $dir/private/rootCA.crt
# For certificate revocation lists.
crlnumber = $dir/crlnumber
crl
               = $dir/crl/intermediate.crl
crl extensions = crl ext
default_crl_days = 30
# SHA-1 is deprecated, so use SHA-2 instead.
            = sha256
default md
cert_opt
               = ca default
               = ca default
default days
               = 375
                = no
preserve
policy
                = policy loose
[ policy strict ]
# The root CA should only sign intermediate certificates that match.
# See the POLICY FORMAT section of `man ca`.
countryName
                      = match
                     = match
stateOrProvinceName
organizationName = match
organizationalUnitName = optional
commonName
                      = supplied
emailAddress
                      = optional
[ policy loose ]
# Allow the intermediate CA to sign a more diverse range of certificates.
# See the POLICY FORMAT section of the `ca` man page.
countryName = optional
                     = optional
stateOrProvinceName
localityName = optional
organizationName = optional
organizationalUnitName = optional
          s = supplied
commonName
emailAddress
[ req ]
# Options for the `req` tool (`man req`).
default bits = 2048
distinguished_name = req_distinguished_name
             = utf8only
string mask
# SHA-1 is deprecated, so use SHA-2 instead.
default md
            = sha256
```

OPTIGA™ TPM Application Note

Integration of TLS Functionality for OPTIGA[™] TPM SLx 9670 TPM 2.0

TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

Extension to add when the -x509 option is used. x509 extensions = v3 ca [req distinguished name] # See https://en.wikipedia.org/wiki/Certificate signing request>. = Country Name (2 letter code) countryName = State or Province Name stateOrProvinceName = Locality Name localityName 0.organizationName = Organization Name organizationalUnitName = Organizational Unit Name commonName = Common Name emailAddress = Email Address # Optionally, specify some defaults. countryName default = US stateOrProvinceName default = California localityName default = Milpitas = Infineon 0.organizationName default organizationalUnitName default = DSS emailAddress default [v3 ca] # Extensions for a typical CA (`man x509v3 config`). subjectKeyIdentifier = hash authorityKeyIdentifier = keyid:always,issuer basicConstraints = critical, CA:true keyUsage = critical, digitalSignature, cRLSign, keyCertSign [v3 intermediate ca] # Extensions for a typical intermediate CA (`man x509v3 config`). subjectKeyIdentifier = hash authorityKeyIdentifier = keyid:always,issuer basicConstraints = critical, CA:true, pathlen:0 keyUsage = critical, digitalSignature, cRLSign, keyCertSign [usr cert] # Extensions for client certificates (`man x509v3 config`). basicConstraints = CA:FALSE nsCertType = client, email nsComment = "OpenSSL Generated Client Certificate" subjectKeyIdentifier = hash authorityKeyIdentifier = keyid,issuer keyUsage = critical, nonRepudiation, digitalSignature, keyEncipherment extendedKeyUsage = clientAuth, emailProtection [server cert] # Extensions for server certificates (`man x509v3 config`). basicConstraints = CA:FALSE nsCertType = server nsComment = "OpenSSL Generated Server Certificate" subjectKeyIdentifier = hash authorityKeyIdentifier = keyid,issuer:always keyUsage = critical, digitalSignature, keyEncipherment extendedKeyUsage = serverAuth [crl ext] # Extension for CRLs (`man x509v3 config`). authorityKeyIdentifier=keyid:always [ocsp] # Extension for OCSP signing certificates (`man ocsp`). basicConstraints = CA:FALSE





TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer
keyUsage = critical, digitalSignature
extendedKeyUsage = critical, OCSPSigning

, 🥝 pi@raspberrypi-os: ~/tpm_hardened_tls			
GNU nano 2.7.4		File: ./pki/openssl.cnf	
<pre>[ocsp] # Extension for OCSP signing certificates basicConstraints = CA:FALSE subjectKeyIdentifier = hash authorityKeyIdentifier = keyid,issuer keyUsage = critical, digitalSignature extendedKeyUsage = critical, OCSPSigning</pre>	(`man ocsp`).		
File Name to Write: ./pki/openssl.cnf			
^G Get Help M-D D	DOS Format	M-A Append	M- <mark>B</mark> Backup File
C Cancel M-M M	1ac Format	M-P Prepend	^⊤ To Files

Figure 6 Copying openssl.cnf reference file

2.5.3 Creating the Root CA and Its Certificate

At the core of the PKI there is the Root CA where the chain of trust originates. In normal practice you would use an established CA like for example GlobalSign [10].

For the purpose of this application note we will use OpenSSL to create a Root Certificate Authority. This is not advised for production purposes.

Within our working directory "tpm_hardened_tls", create an RSA key pair.

Use OpenSSL to create the Root CA Key pair.

Code Listing 5 Create RootCA key pair

001 openssl genrsa -out ./pki/private/rootCA.key 2048

Creating a Self-Signed RootCA Certificate

Code Listing 6 Self-Signed RootCA Certificate

```
001 openssl req -config ./pki/openssl.cnf -key
   ./pki/private/rootCA.key -new -x509 -days 7300 -sha256 -extensions
   v3_ca -out ./pki/private/rootCA.crt
   002 echo 1000 > ./pki/serial
```



TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

Enter the Root CA certificate information as shown in Figure 7.

pi@raspberrypi-os:-/tpm hardened_tls \$ openssl req -config ./pki/openssl.cnf -key ./pki/private/rootCA.key -new -x509 -days 7300 -sha256 -extensions v3_ca -out ./pki/private/rootCA.crt
into your certificate request.
what you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
---Country Name (2 letter code) [US]:US
State or Province Name [California]:California
coality Name [Miptitas]:Wilpitas
Organizational Unit Name [DS]:DSS
Common Name [I:www.infireon.com
Email Address []:

Figure 7 Root CA Distinguished Name or DN

Reading our RootCA

Code Listing 7

001 openssl x509 -in ./pki/private/rootCA.crt -noout -text

OPTIGA[™] TPM Application Note

Integration of TLS Functionality for OPTIGA[™] TPM SLx 9670 TPM 2.0



TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

<pre>pi@raspberrypi-os:~/tpm_hardened_tls \$ openssl x509 -in ./pki/private/rootCA.crt -noout -text Contificate:</pre>
Version: 3 (0v2)
Serial Number
$f_1 + f_2 + f_3 $
Signature Algorithm: sha256WithRSAEncryntion
TSUER: C = US, ST = California, L = Milnitas, O = Infineon, OL = DSS, CN = WWW.infineon.com
Validity
Not Before: Aug 16 04:10:25 2019 GMT
Not After : Aug 11 04:10:25 2039 GMT
Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:b0:7b:dd:d1:34:71:0b:e3:18:26:09:54:9c:04:
b9:5e:15:a3:cf:64:da:8c:62:86:57:5a:80:0c:fe:
7f:c9:35:a1:f9:2a:85:a4:89:df:fc:d0:15:5b:e5:
31:24:be:e7:b7:bb:43:95:0e:a4:3e:43:90:11:6c:
e6:5c:e5:91:b7:b1:15:86:25:1a:cc:50:50:f3:cf:
73:64:d2:2d:d3:38:01:a6:31:26:5e:cf:1f:b7:92:
bc:47:b9:d0:7d:c9:d5:a4:26:3f:eb:6b:03:03:21:
d4:65:7f:14:40:ed:2b:6e:ac:af:6c:70:36:ba:56:
47:b0:d7:95:eb:36:5e:a2:0c:5a:03:41:04:c6:97:
61:ba:22:85:cb:1f:1e:20:76:6c:78:fc:82:79:64:
1e:+6:06:39:9b:8+:+2:00:94:e3:7d:62:+3:78:9b:
+7:23:19:98:62:7b:2b:77:60:ca:0c:e4:+d:de:59:
4+:b4:d3:ed:59:b+:3+:95:a+:+d:6c:+1:e/:51:5/:
ee:td:tb:85:/9:a8:1b:50:39:2e:62:e/:1d:b9:86:
10:2T:/0:94:9C:52:DT:4C:6D:ae:99:e3:D/:48:8/:
d1:4/:01:8d:10:55:01:59:44:d0:D1:00:50:d1:40:
bd.13
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Subject Kev Identifier:
71:0D:F7:1F:51:3C:AC:12:6C:8F:66:77:7A:99:9A:25:47:BC:96:EA
X509v3 Authority Kev Identifier:
keyid:71:0D:F7:1F:51:3C:AC:12:6C:8F:66:77:7A:99:9A:25:47:BC:96:EA
X509v3 Basic Constraints: critical
CA: TRUE
X509v3 Key Usage: critical
Digital Signature, Certificate Sign, CRL Sign
Signature Algorithm: sha256WithRSAEncryption
9c:d5:5f:10:43:09:a1:c4:6f:25:0d:69:fe:86:b4:2e:06:ac:
8f:5f:49:b2:14:84:c4:47:d4:2e:fb:89:03:d2:84:1a:3d:3b:
58:53:b0:9d:5c:e4:b7:60:96:de:c5:ee:2f:64:71:31:d1:93:
9b:2f:e8:48:03:3b:31:0a:3f:b3:02:e4:9b:15:70:b8:dd:c7:
81:dt:a0:98:td:7a:0a:05:20:08:e0:25:4a:d7:d8:a9:5e:5d:
b1:c0:5b:/f:18:bc:a4:94:ea:b8:41:8e:b1:/5:4b:e5:5e:a1:
95:40:80:24:00:40:T7:10:95:29:58:T2:80:09:32:08:80:T1:
T4:08:T6:23:40:e2:64:10:05:7a:2a:9a:0T:33:80:38:19:43: 0d:2d:72:ef:fe:f0:b0:26:02:70:be:10:28:2b:0e:de:61:d6:
bh fd ad ad a a a a a a a a a a a a a a a
h3+29+2f+95+2f+d2+52+97+84+8f+5h+f6+3c+26+28+d7+35+b2+
43:0a:3e:a1:49:3c:95:e7:6d:ab:92:e1:16:ab:25:f1:c1:d0:
46:46:ef:68:b8:d9:2e:84:e1:48:4c:1b:a2:97:1c:02:36:64:
e9:7e:ba:07:70:6a:e9:ea:af:6b:dc:27:13:09:60:77:c3:c7:
5b:28:b2:42

Figure 8 Root CA Certificate



TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

2.5.4 Creating the Intermediate CA and Its Certificate

From a security perspective it is always advised to create an intermediate CA signed by the Root CA.

Create the Intermediate CA Key Pair.

Code Listing 8

001 openssl genrsa -out ./pki/private/intCA.key 2048

Create a CSR for the Intermediate CA

Code Listing 9

```
001 openssl req -config ./pki/openssl.cnf -extensions
v3_intermediate_ca -new -sha256 -key ./pki/private/intCA.key -out
./pki/csr/intCA.csr
```

Enter the CSR information as shown in Figure 9.

```
pi@raspbervpi-os:~/tpm_hardened_tls $ mkdir ./pki/csr
pi@raspbervpi-os:~/tpm_hardened_tls $ openssl req -config ./pki/openssl.cnf -extensions v3_intermediate_ca -new -sha256 -key ./pki/private/intCA.key -out ./pki/csr/intCA.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:US
State or Province Name [California]:California
Locality Name [Milpitas]:Milpitas
Organization Name [Infineon]:Infineon
Organizational Unit Name [DS5]:DSS
Common Name []:www.infineon.com
Email Address []:
```

Figure 9 Creating Intermediate CA CSR

Create the Intermediate CA certificate

Code Listing 10

```
001 openssl ca -config ./pki/openssl.cnf -extensions
v3_intermediate_ca -days 3650 -notext -md sha256 -in
./pki/csr/intCA.csr -out ./pki/private/intCA.crt
```



TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

pi@raspberrypi-os:~/tpm_hardened_tls \$ openssl ca -config ./pki/openssl.cnf -extensions v3_intermediate_ca -days 3650 -notext -md sha256 -in ./pki/csr/intCA.csr -out ./pki/private/intCA
Jsing configuration from ./pki/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
Serial Number: 4097 (0x1001)
Validity
Not Before: Aug 16 05:34:58 2019 GMT
Not After : Aug 13 05:34:58 2029 GMT
Subject:
countryName = US
stateOrProvinceName = California
localityName = Milpitas
organizationName = Infineon
organizationalUnitName = DSS
commonName = www.infineon.com
X509v3 extensions:
X509v3 Subject Key Identifier:
9D:C3:99:85:A8:AA:33:8A:9A:34:DA:F8:57:01:A5:BD:08:7E:51:3E
X509v3 Authority Key Identifier:
keyid:5D:AC:35:6A:29:96:CD:65:3E:2F:ED:3E:67:F2:19:57:FC:F6:06:60
X509v3 Basic Constraints: critical
CA:TRUE, pathlen:0
X509v3 Key Usage: critical
Digital Signature, Certificate Sign, CRL Sign
Certificate is to be certified until Aug 13 05:34:58 2029 GMT (3650 days)
Sign the certificate? [y/n]:y
l out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Jata Base Updated



Verify Signature process

Code Listing 11

```
001 openssl verify -verbose -x509_strict -CAfile
./pki/private/rootCA.crt ./pki/private/intCA.crt
```

pi@raspberrypi-os:~/tpm_hardened_tls \$ openssl verify -verbose -x509_strict -CAfile ./pki/private/rootCA.crt ./pki/private/intCA.crt ./pki/private/intCA.crt: OK

Figure 11 Verify Intermediate CA vs. Root CA

Reading our Intermediate CA Certificate

Code Listing 12

001 openssl x509 -in ./pki/private/intCA.crt -noout -text

OPTIGA[™] TPM Application Note

Integration of TLS Functionality for OPTIGA[™] TPM SLx 9670 TPM 2.0



TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

pi@raspberrypi-os:~/tpm_hardened_tls \$ openssl x509 -in ./pki/private/intCA.crt -noout -text
Certificate:
Data:
Serial Number: 4097 (0x1001)
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com
Validity
Not Before: Aug 16 05:34:58 2019 GMT
Not After : Aug 13 05:34:58 2029 GMI
Subject: C = OS, ST = California, C = Milpitas, O = Infineon, OO = DSS, CN = WWW.Infineon.com Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:b1:34:90:e2:cb:40:13:b1:ff:8c:ee:37:76:c7:
b0:b3:81:72:03:d7:27:a4:f2:50:9f:da:96:19:b6:
b8:de:86:12:19:25:4b:b0:ab:70:70:f0:69:de:79:
69:ca:d3:40:12:32:dc:05:21:20:97:69:cD:de:5e: 2d:f1:31:d2:o2:86:o4:02:co:cf:06:92:62:b4:1o:
e8:9f:fb:e4:81:df:c4:03:b7:cf:e4:a0:81:69:8e:
d9:f0:95:fa:03:0f:9d:2d:7b:4b:fe:44:94:7f:3a:
44:8f:d8:62:4e:e5:20:67:04:24:54:dc:88:2f:13:
51:03:dd:44:bb:39:c5:f4:c1:ab:00:db:36:81:22:
a1:5d:dd:78:a6:4e:59:c6:f1:63:77:5b:e1:58:11:
ad:a2:b3:3f:9a:c8:dd:01:80:6e:e0:d5:9f:61:7e:
ba:3c:ce:/8:35:c3:/b:t0:e5:49:/t:90:6e:c2:96:
22:72:02:40:11:41:42:10:e0:e0:00:44::00:94: 2e:e1:41:e1:h5:fa:6f:59:67:3e:hd:2e:25:7c:13:
f3:9d:d2:bb:ee:78:a9:c5:c0:56:ac:d1:e5:c2:97:
d3:5a:0d:42:d1:9b:41:f1:3d:dc:3c:e8:c6:b0:ac:
9a:96:1c:ca:8e:9b:fb:90:1a:cd:6c:63:c9:da:d4: 44:77
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Subject Key Identifier:
9D:C3:99:85:A8:AA:33:8A:9A:34:DA:F8:57:01:A5:BD:0B:7E:51:3E
kevid:5D:AC:35:6A:29:96:CD:65:3E:2E:ED:3E:67:E2:19:57:EC:E6:06:60
X509v3 Basic Constraints: critical
CA:TRUE, pathlen:0
X509v3 Key Usage: critical
Digital Signature, Certificate Sign, CKL Sign
35:88:b4:06:32:b9:e4:86:7d:8f:6e:c5:79:4a:33:89:23:96:
a7:f0:fe:93:4e:a3:ee:ad:18:47:26:c7:28:29:d8:11:51:7c:
79:47:79:d3:c9:c2:2b:cb:2f:b2:cd:8d:7f:89:e7:97:e7:a4:
eb:c1:b6:1c:fc:f1:9a:05:50:9f:18:4a:5f:5c:34:e5:37:7d:
91:0c:5c:a7:9e:1f:f4:db:0d:2e:d7:6b:2c:01:d1:37:4a:86:
87:c6:2b:be:b9:6e:18:16:58:f5:a2:90:f1:58:d8:f0:fa:cf:
e9:a0:0/:TC:02:09:aT:99:00:71:60:3D:95:T3:6T:C0:3D:eD: 0f:2d:1a:5d:10:10:h6:02:1d:62:56:0c:25:0a:aa:f7:16:06:
ad:fd:d8:6f:23:32:a3:20:e1:8d:5f:fe:32:87:69:a7:81:
12:c7:45:94:a9:fb:78:b5:df:03:04:4e:d2:4c:dd:4d:d8:8f:
37:fc:7f:fb:62:38:22:80:14:b0:a0:38:ca:28:14:c0:cc:9b:
17:37:80:bb:2c:44:87:f5:2f:9e:d3:cf:d6:ed:72:b5:dc:64:
3e:b0:d7:7f:3b:84:0c:57:f7:69:7a:04:f5:95:b3:7a:9e:3f:
52:4a:52:5c:27:c8:b3:a3:9a:57:7a:28:5d:6e:ac:a2:7d:78:
16:6/:18:76

Figure 12 Intermediate CA Certificate

Note:

As mentioned before, the exercise of creating the Root CA and Intermediate CA using OpenSSL are for demonstration purposes only. In real applications these would be managed by the Certificate Authority like for example GlobalSign [10].

TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0



2.5.5 Creating Client/Endpoint Key Pair Using SLx 9670 TPM2.0

Use SLx 9670 TPM2.0 OpenSSL engine to create a TPM 2.0 key pair for the Client/Endpoint.

Note:

As explained in Section 2.5.1, the TPM wraps (encrypts) the private key and stores it outside the TPM. The encrypted key blob [Figure 13] is encapsulated between "-----BEGIN TSS2 PRIVATE KEY-----" and "-----END TSS2 PRIVATE KEY-----"

Create the Client/Endpoint key pair with Password Security Policy using SLx 9670 TPM2.0. Set the password to "abc". This is only for demonstration purposes. *DO NOT USE FOR PRODUCTION*.

Code Listing 13

001 tpm2tss-genkey -a rsa -s 2048 ./tpm2/tpm_wrapped_keys/client.key -p abc

pi@raspberrypi-os:~/tpm_hardened_tls \$ cat ./tpm2/tpm_wrapped_keys/client.key -----BEGIN TSS2 PRIVATE KEY-----MIICDwYGZ4EFCgEDoAMBAQACAQAEggEYARYAAQALAAYEcgAAABAAEAgAAAEAAQEA 9DaEJScgvUX/b6L7kk8Z3AK2QrTpoRTsaMdtOROXi7FvwpTLDKBTiy1p0KcTLUFO y+RZoUIFA6vMM1PJJjAtnbUfw0oRaQOaNes0bSeyKoutxvoqC14013dRNrnfq31n qOqWIyeJDsqsH0drotLU53IvNS4d5U+2nxhYSqG13f+mKQrX4AasvpC706uFk5W4 QUwAN/d7U2jaU/rg6+sV6Pq1/ua+XNhKWwFkZmcEJ02k3yFPqaRXZszZ115jDymQ 5ra34gokVZBuTZuBU1zmsR1kZAT5fn58mJ46pG1GsdrfPBDsbh+TYfIg1FI5hwTz u6cneTHM1MpaaXqOdpwWbQSB4ADeACBBKhT+vB+8vBtEfYALRHt3V7VEfq9aIEKp zBs1XRXYEgAQ2nKc+4RppWUyKz1IhXabmu+pF1DYShqXVR6x8Nd7yN0oN0kLj6gf 9T4xhSqXMzd510US6FsoSbSDdYxT/h0iQA3i/j0SM+aDsLxKPSUi3AEyDu7JihCA pSQd/JXViydoFohdCq1nkzb/etF1mJF56UiLIiJnptpZiD8QuprVwW7w+aK9ZTz6 WWqTeX9DazMX+tm11cL1EFUrg4Vp1JJ3reZMykbbSmjKQd2Ab36bgq72W8uF1H7G jHIY -----END TSS2 PRIVATE KEY-----

Figure 13 Client/Endpoint Key Pair wrapped by TPM

2.5.6 Creating Client/Endpoint CSR Using SLx 9670 TPM2.0

Create the Client/Endpoint CSR using SLx 9670 TPM2.0. The key pair we just created will be managed by the SLx 9670 TPM2.0.

Note: Remember that the private key is wrapped outside TPM 2.0 and will be loaded in order to be used for this purpose. Also, the key pair has been configured with a security policy that requires a password for the key to be used.

Code Listing 14

```
001 openssl req -keyform engine -engine tpm2tss -config
./pki/openssl.cnf -key ./tpm2/tpm_wrapped_keys/client.key -new -out
./tpm2/csr/client.csr
```



Note: When asked for the password, input the set password "abc".

Fill the Certificate Request Information as shown in Figure 14.

pi@raspberrypi-os:-/tpm_hardened_tls \$ openssl req -keyform engine -engine tpm2tss -config ./pki/openssl.cnf -key ./tpm2/tpm_wrapped_keys/client.key -new -out ./tpm2/csr/client.csr engine "tpm2tss" set. Enter password for user key: You are about to be asked to enter information that will be incorporated into your certificate request. What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank For some fields there will be a default value, If you enter '.', the field will be left blank. -----Country Name (2 letter code) [US]:US State or Province Name [California]:California Locality Name [Infineon]:Infineon Organization Name [Infineon]:Infineon Organization Name [DSS]:DSS Common Name []:www.infineon.com Email Address []:

Figure 14 Client/Endpoint CSR Information

Read the Client/Endpoint CSR

Code Listing 15

001 openssl req -in ./tpm2/csr/client.csr -noout -text

ni@	antennumi can (tam bandered the & energian in (tam)(can (client can prout taut
pi@ra	<pre>spperrypl-os://tpm_nardened_tis > openssi req -in ./tpm2/csr/client.csr -noout -text</pre>
Centi	itate Request:
6	Version: 1 (AvA)
	Subject: C = US, ST = California, L = Milnitas, O = Infineon, OU = DSS, CN = www.infineon.com
	Subject Public Key Info:
	Public Key Algorithm: rsaEncryption
	Public-Key: (2048 bit)
	Modulus:
	00:f4:36:84:25:27:20:bd:45:ff:6f:a2:fb:92:4f:
	19:dc:02:b6:42:b4:e9:a1:14:ec:68:c7:6d:39:13:
	97:8b:b1:6f:c2:94:cb:0c:a0:53:8b:2d:69:d0:a7:
	13:2d:41:4e:cb:e4:59:a1:42:1f:03:ab:cc:32:53:
	c9:26:30:2d:9d:b5:1f:c3:4a:11:69:03:9a:35:eb:
	34:6d:27:b2:2a:8b:ad:c6:fa:2a:0b:5e:34:d7:77:
	51:36:b9:df:ab:79:67:a8:ea:96:23:27:89:0e:ca:
	ac:1f:47:6b:a2:d2:d4:e7:72:2f:35:2e:1d:e5:4f:
	b6:9f:18:58:4a:a1:a5:dd:ff:a6:29:0a:d7:e0:06:
	ac:be:90:bb:3b:ab:85:93:95:b8:41:4c:00:37:f7:
	7b:53:68:da:53:fa:e0:eb:eb:15:e8:fa:a5:fe:e6:
	be:5c:d8:4a:5b:01:64:66:67:04:27:4d:a4:df:21:
	4f:a9:a4:57:66:cc:d9:d6:5e:63:0f:29:90:e6:b6:
	b7:e2:0a:24:55:90:6e:4d:9b:81:53:5c:e6:b1:19:
	64:64:04:f9:7e:7e:98:9e:3a:a4:69:46:b1:da:
	df:3c:10:ec:6e:1f:93:61:f2:20:d4:52:39:87:04:
	f3:bb:a7:27:79:31:cc:d4:ca:5a:69:7a:8e:76:9c:
	16:6d
	Exponent: 65537 (0x10001)
	Attributes:
	a0:00
S	ignature Algorithm: sha256WithRSAEncryption
	10:11:35:51:45:d9:ae:2d:3e:ae:ea:e9:69:b0:03:06:8/:c2:
	b9:51:22:42:49:99:70:b8:d3:22:e2:d3:cd:91:71:C3:
	48:3a:3/:5a:Ca:43:2d:a9:ce:D5:Ca:05:cb:c0:t+:a7:5b:ac:
	5d:32:e9:7e:Tb:28:95:d1:52:1f:4f:73:d0:68:T9:85:e0:40:
	1/:a0:0/:1c:ac:t/:1/:20:1t:09:20:94:80:/a:aa:t/:52:00:
	481T21051821aT1T17/a12e1aa10a10a10015014T12a1e01C01TT1
	39:a5:a4:b0:27:05:50:80:a5:16:a5:16:89:23:47:a6:75:54:
	6/1751e0:cc:3818/:6/2cT:23:4/18/:01:23:cD18C:08174:99:
	04:99:88:25:70:7/:22:06:09:00:41:35:09:02:70:10:00: 16:06:04:42:00:00:00:06:16:44:06:04:04:16:10:04:02:00:00:00:00:00:00:00:00:00:00:00:00:
	10:00:40:42:40:05:51:10:04:00:40:04:00:10:122:84:29: d:d:52:51:21:20:12:05:11:20:02:1d:100:1d:01:20:16:0:04:04:04:00:100:100:100:100:100:100:1
	uu,u), 51, 55, 68, 51, 46, 51, 53, 50, 20, 20, 20, 20, 73, 52, 74, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50
	o.eu./o.uou.eu.ou.eu.eu.eu.eu.eu.eu.eu.eu.eu.eu.eu.eu.eu
	10.74.50.51.11.50.10.15.12.103.01.00.125.121.17.12.122.120.144:
	20.75.70.05.02.10.12.70.15:70:27.59:47:00:17:75:22:00: 00.06.75.46
	J01J0126170



TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

Figure 15 Client/Endpoint CSR

2.5.7 Signing Client/Endpoint CSR with RootCA

Use the Intermediate CA to sign the created Client/Endpoint CSR

Code Listing 16

```
001 openssl x509 -req -days 365 -in ./tpm2/csr/client.csr -CA
./pki/private/intCA.crt -CAkey ./pki/private/intCA.key -CAcreateserial
-out ./tpm2/certs/client.crt
```

pi@raspberrypi-os:~/tpm_hardened_tls \$ openssl x509 -req -days 365 -in ./tpm2/csr/client.csr -CA ./pki/private/intCA.crt -CAkey ./pki/private/intCA.key -CAcreateserial -out ./tpm2/certs/client.crt Signature ok subject=C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Getting CA Private Key

Figure 16 Signing Client CSR with Intermediate CA

To verify the certificate chain, we use the parameter -untrusted for the Intermediate CA certificate file. This is the parameter OpenSSL has assigned for this verification.

Code Listing 17

001 openssl verify -CAfile ./pki/private/rootCA.crt -untrusted ./pki/private/intCA.crt ./tpm2/certs/client.crt

pi@raspberrypi-os:~/tpm_hardened_tls \$ openssl verify -CAfile ./pki/private/rootCA.crt -untrusted ./pki/private/intCA.crt ./tpm2/certs/client.crt ./tpm2/certs/client.crt: OK

Figure 17 Client Certificate Chain verification

Read the Client/Endpoint Certificate

Code Listing 18

001 openssl x509 -in ./tpm2/certs/client.crt -noout -text



TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

<pre>pigerspberrypi-os:-/tpm_handemsd_tls \$ openssl x509 -in ./tpm2/certs/client.crt -noout -text Cortificate: Data:</pre>		
<pre>Certificate: Data: Version: 1 (0x0) Serial Number: ae:cb:a0:21:a1:9d:43:59 Signature Algorithm: sha250withRSAEncryption Issuer: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Validity Not Sefrore: Aug 16 06:00:29 2020 0MT Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, C = US, ST = US</pre>	pi@raspberrypi-os:~/tpm_hardened_tls \$ openssl x509 -in ./tpm2/certs/client.crt -noout -text	
<pre>Data: Wersion: 1 (0x0) Serial Number: ae:cb:a0:21:a1:9d:43:59 Signature Algorithm: sha256WithRSAEncryption Insuer: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Validity Not Before: Aug 16 06:00:29 2019 GMT Not After : Aug 15 06:00:29 2020 GMT Public Key Info: Public Key Info: Public Key Info: Public Key (2048 bit) Modulus:</pre>	Certificate:	
<pre>Version: 1 (xxe) Serial Number: ae:0:a80:21:a1:90:43:59 Signature Algorithm: sha256withKSAEncryption Issuer: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Walidity Not Refore: Aug 15 06:00:29 2020 GMT Not After: Aug 15 06:00:29 2020 GMT Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, C = US, ST = California, L = Milpitas, ST = US, ST = U</pre>	Data:	
<pre>Serial Number: ac:cb:ab:21:al:93:43:59 Signature Algorithm: sha25&WithKSAEncryption Ifsuer: C = US, S = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Walidity Not Before: Aug 16 06:00:29 2020 GMT Not After : Aug 15 06:00:29 2020 GMT Not After : Aug 15 06:00:29 2020 GMT Diverse (C = US, S T = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject Public Key Info: Public Key Info: Public Key Info: Nodulas: 00:f4:36:084:25:27:20:bd:45:ff:6f:a2:fb:92:4f: 19:dc:20:b6:42:b4:ed:3a1:41:4e:c68:c7:6d:39:13: 97:8b:b1:6f:c2:94:cb:06:a0:53:08:24:69:00:a7: 13:2d:14:4e:cb:4e:95:a1:42:16:30:8b:23:53: c6:26:30:24:9d:b5:1f:c3:34:11:69:03:9a:33:eb: 34:6d:77:b2:2a:8b:dd:6f:a0:77:77: 51:36:b9:df:ab:79:67:a38:ea:96:23:27:89:00:ca: ac:16:79:65:38:43:a1:a5:dd:ff:a6:29:00:47:e0:06: ac:bb:99:bb:3b:a5:59:39:5b:41:4:c0:07:77: 77:55:68:da:53:fa:e0:be:bb:15:e0:fa:a5:fe:e6: be:55:c1:d4:45:56:14:27:42:24:48:77:22:14: 4f:a9:a4:57:66:cc:d9:d6:5e:63:0f:29:99:0e:6b: 17:e2:0a:24:55:99:0b:66:44:d9:47:27:44:a4:7:e1: 4f:a9:a4:57:66:cc:d9:d6:5e:63:0f:29:99:0e:6b: 17:e2:0a:24:55:99:0b:66:44:d9:47:24:d4:27:24:24: 4f:a9:a4:57:66:cc:d9:d6:5e:63:0f:29:99:0e:6b: 17:e2:0a:24:55:99:0b:66:44:04:6f:29:13:35:c1:df:20:04:25:23:27:04: 16:65 Exponent: 6537 (0x10001) Signature Algorithm: sha256WithKSAEncryption 62:cf:5f:ef:db:0b:a3:17:e8:31:80:16:23:69:7a:8e:76:9c: 16:65 Exponent: 6537 (0x10001) Signature Algorithm: sha256WithKSAEncryption 62:cf:5f:fe:fb:0b:83:19:30:16:23:18:07:60:24:35:19:19:16:24:45:23:25:24:45:23:23:24:45:23:25:24:45:23:25:24:45:23:25:24:45:23:25:24:45:23:25:24:45:23:25:24:45:23:25:24:45:23:25:24:45:23:25:24:45:24:24:45:22:25:24:45:23:24:24:45:22:25:24:45:23:24:24:24:24:24:24:24:24:24:24:24:24:24:</pre>	Version: I (0x0)	
<pre>latercial/141/941/94/94/94 Signature Algorithm: sha25WithKSAEncryption Issuer: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Walidity Not Before: Aug 16 06:00:29 2020 GNT Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: Aug 15 06:00:29 2020 GNT Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject: Aug 16 06:00:29 2020 GNT Modulus: 00:0f4:36:84:25:27:20:bd:45:ff:6f:a2:fb:92:4f: 19:dc:02:06:42:bd:06:20:85:38:02:d6:06:00:a7: 13:2d:41:4e:cb:e4:59:a1:14:ec:68:c7:6d:39:13: 01:26:10:16:16:21:24:8b:16:16:29:80:24:08:a7: 13:2d:41:4e:cb:e4:59:a1:42:1f:08:ab:cc:32:53: 01:26:10:46:16:16:17:67:38:ea:96:13:8b:24:69:08:a7: 13:2d:41:4e:cb:e4:59:a1:42:1f:08:ab:cc:32:53: 01:26:10:46:16:10:79:67:38:ea:96:13:21:78:90:Be:ca: ac:16:47:06:16:16:16:26:23:27:89:Be:ca: ac:16:47:06:16:16:16:26:23:27:78:90:Be:ca: ac:16:49:06:19:30:ab:85:93:95:58:41:4c:00:37:f7: 7b:55:68:d6:33:fa:e0:eb:eb:15:68:fa:25:90:e6:66: bb:55:06:10:10:16:46:66:70:10:47:74:46:12:11: 4f:a5:a4:57:76:16:46:65:16:10:10:16:16:16:10:10:10:10:10:10:10:10:10:10:10:10:10:</pre>	Serial Number:	
<pre>signature Algorithm: Sind Source Algorithm All of Parks Insure: C = US, S = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Validity Not After : Aug 15 06:00:29 2020 GMT Not After : Aug 15 06:00:29 2020 GMT Not After : Aug 15 06:00:29 2020 GMT Public Key Info: Public Key Info: Public Key Info: Public Key Info: Nodula: 00:f4:36:84:25:27:20:bdt45:ff:6f:a2:fb:92:4f: 00:f4:36:84:25:27:20:bdt45:ff:6f:a2:fb:92:4f: 00:f4:36:84:25:27:20:bdt45:ff:6f:a2:fb:92:4f: 00:f4:36:84:25:27:20:bdt45:ff:6f:a2:fb:92:4f: 00:f4:36:84:25:27:20:bdt45:ff:6f:a2:fb:25:25: C = 226:30:24:9dt:bd:14:ed:a1:14:ed:80:76:dd:39:13: 97:8b:b1:6f:c1:94:cb:84:75:23:25:25: C = 226:30:24:9dt:b5:1f:c1:4a:11:69:80:39:35:eb: 34:6d:27:b2:2a:8b:ad:c6:fa:22:adt:b5:24:c0:777: 51:36:b9:df:ba:7:0f:7a:8a:ed:25:22:25:80:ec:aa: ac:1f:47:06:ba:2:d2:d4:eb:777:27:55:2e:1d:e5:4f: b6:9f:18:88:4a:a1:a5:dd:ff:22:98:ad7:e0:e0: ac:be:99:0b:3b:ab:85:93:25:5b:84:14:c0:80:37:f7: 7b:55:66:dd:35:fa:e0:eb:eb:15:ed:fa:25:64:75:90:ed:16: b6:5c:d8:4a:35:01:64:d5:0f:12:29:e0:61: b6:5c:d8:4a:35:01:64:d5:0f:12:29:e0:61: b6:7:e2:4a:24:55:90:6e:4d:9b:81:53:5c:e6:b1:19: 64:64:04:f9:7e:77:79:30:e6:4d:35:fa:23:38:70:e0:6b: b7:e2:4a:24:55:90:6e:4d:9b:81:53:5c:e6:b1:19: 64:64:04:f9:7e:77:79:30:e6:4d:95:71:e6:16: b6:5c:d8:4a:35:01:fa:20:6d:17:20:20:d4:52:39:87:e0: b6:16:d1:57:77:71:31:cc:d4:ca:sa:69:73:88:76:9c: b6:6d Exponent: e5537 (0xL0001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:fe:7:0b:3a:3b:9d:4d:e6:c3:31:80:71:77:71: b5:16:d2:10:ec:c6:16:31:72:e2:80:35:d1: b5:6f:e3:15:61:31:72:e3:35:65:20:15:20:61:20:20:de: 53:39:40:db:13:31:72:e3:33:60:65:29:33:5d1: b5:6f:e3:0b:ed:d5:16:67:14:22:80:47:12:20:4f:82:94: d5:16:19:30:ec:d5:16:37:2a:85:36:65:20:35:61: b5:6f:e3:0b:ed:d5:16:67:14:22:80:47:14:27:14:82:74:41:41:40:41:67:20:20:4f:82:34: d7:6f:5f:fe:7:0b:8d:8d:15:77:2a:35:23:86:26:65:90:33:5d1: b5:6f:e3:0b:ed:d5:16:13:72:a0:85:31:80:42:40:20:20:6f: b5:6f:e3:0b:ed:d5:16:13:72:a0:85:31:80:42:40:20:20:6f: b5:6f:e3:70:6f:e3:0b:9f:ed:ed:d5:16:6f:20:80:20:6f:20:</pre>		
<pre>Validity Not Before: Aug 16 06:00:29 2019 GMT Not After: Aug 15 06:00:29 2019 GMT Subject: C = US, ST = California, L = Milpitas, 0 = Infineon, 0U = DSS, CN = www.infineon.com Subject: C = US, ST = California, L = Milpitas, 0 = Infineon, 0U = DSS, CN = www.infineon.com Subject: Public Key Info: Public Key Algorithm: rsaEncryption Public-Key: (2048 bit) Modulus: 00:f4:36:84:25:27:20:bd:45:ff:6f:a2:fb:92:4f: 19:dc:02:b6:42:b4:09:a1:14:ec:68:C7:6d:39:13: 07:8b.b1:6f:c2:94:cb:02:a0:30:00:a0:a7: 13:2d:41:4e:cb:e4:59:a1:42:ef:83:8b:2d:00:a7: 13:2d:41:4e:cb:e4:59:a1:42:ef:83:8b:2d:00:a7: 13:2d:41:4e:cb:e4:59:a1:42:ef:83:8b:2d:00:a7: 13:2d:41:4e:cb:e4:59:a1:42:ef:83:8b:2d:00:a7: 13:2d:41:4e:cb:e4:59:a1:42:ef:83:8b:2d:00:a7: 13:2d:41:4e:cb:e4:59:a1:42:ef:83:8b:2d:00:a7: 13:2d:41:f6:a2:fb:16:6:a2:fb:02:a3:5e:ed: 05:61:61:62:7b:18:58:4a:a1:63:0d:ff:a2:ef:00:37:f7: 51:36:16:61:51:62:32:72:26:35:ef:2d:00:37:f7: 7b:53:61:da:53:71:60:ef:40:93:16:88:f1:a6:20:eb:11:91: 64:64:04:f9:7e:7e:7e:99:99:3a:a4:83:36:ef:e6: be:55:c8:4a:5b:00:64:40:9b:81:33:51:c6:eb:11:91: 64:64:04:f9:7e:7e:7e:99:99:3a:a4:80:46:b1:da: df:3c:10:ec:66:1f:93:61:f2:20:ed:45:87:98:7e:12: 4f:a9:a4:57:66:cc:09:d6:5e:63:67:29:90:eb:2b: 16:6d Exponent: 65:57 (0x10001) Signature Algorithm: sha256WIthRSAEncryption 62:cf:f5:fe:fb:0b:35:9b:fb:44:7:21:fe:44:64:a7:91:7d:7b: 53:fa:3d:66:52:7d:1a:27:48:49:3c:bd:62:33:80:7c:b2:26:de: 53:94:66:db:23:7d:1a:27:48:49:3c:bd:62:33:80:7c:b2:26:de: 53:94:66:db:23:7d:1a:27:48:49:3c:bd:62:33:80:76:1b:35:4a: df:fa:db:fa:4b:fb:46:f5:7a:a3:80:f5:1a:35:de:46:1a:a3:a3:a4:f5:1a:35:a4:a5:a4:f5:1a:35:a4:a5:a4:f5:1a:35:a4:a5:a4:f5:1a:35:a4:a5:a4:55:27:61:a5:35:62:a5:35:d1: b5:6f:ef:fb:0b:85:7d:1a:37:2a:65:21:35:ef:61:35:44: df:fa:db:fa:4d:6f:7a:d7:2a:38:87:7a:82:35:a4:a5:a5:35:d1: b5:6f:ef:7b:0F:85:9b:fb:48:72:fe:4a:64:79:1C:2:26:de: 53:94:96:eb:23:7d:1a:27:48:49:3c:bd:62:33:80:7:62:25:de: 53:94:96:eb:23:7d:1a:27:48:49:3c:bd:62:33:80:75:61:35:35:d1: b5:6f:ef:7b:ef:8b:db:66:97:2a:33:80:75:61:35:35:d1: b5:6f:ef:7b:75:8b:db:65:97:2a:38:85:</pre>	Signature Algorithm: ShazoowithSaencryption	
<pre>Windly Not Before: Aug 16 06:00:29 2019 GMT Not After : Aug 15 06:00:29 2020 GMT Subject Public Key Info: Public Key JGT California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject Public Key Info: Public Key Algorithm: rsaEncryption Public Key Lot 2005 Content of the set of the set</pre>	$V_{aldiffu}$	
<pre>Not After : Aug 15 06:00:29 2020 GMT Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject Public Key Info: Public-Key: (2048 bit) Modulus:</pre>	Not Refore: Aug 16 06:00:20 2010 GMT	
<pre>Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com Subject Public Key Info: Public Key Ugorithm: rsaEncryption Public Key (2048 bit) Modulus: 00:f4:36:84:25:27:20:bd:45:ff:6f:a2:fb:92:4f: 19:dc:22:b6:42:bd:e0:a1:14:ec:68:c7:6d:39:13: 97:8b:bl:6f:c2:94:cb:0c:a0:53:8b:2d:90:0d:a7: 13:2d:41:4e:cb:e4:59:a1:42:1f:03:ab:cc:32:53: c9:26:30:2d:9d:0b:17:c3:4a:11:69:03:9a:35:eb: 34:6d:77:b2:2a:8b:ad:c6:12:24:33:d0;77: 51:36:b9:df:a0:79:67:a8:ea:96:32:27:89:06:ca: ac:16:97:0b:13:58:44:a1:a5:dd:ff:a6:29:0a:d7:e0:06: ac:be:90:bb:3b:ab:83:39:95:b9:41:42:00:37:77: 7b:53:66:4d:55:13:4a:eb:15:8e:ff:a5:fe:e6: be:5c:d8:4a:5b:01:64:66:67:04:27:44:44:df:21: 4f:a3:45:57:66:cd:36:65:63:07:29:29:0e:6b: b7:e2:8a:24:55:90:6e:4d:9b:81:35:57:0e:6b:19: 64:64:46:19:7e:7e:7c:98:99:66:4b:19: 64:64:46:19:7e:7e:7c:98:99:66:4b: b6:37:7e:13:61:46:55:97:81:aa:66:51:96:29:0e:6b: b7:e2:8a:24:55:90:6e:4d:9b:81:53:57:0e:6b:19: 64:64:46:19:7e:7e:7c:98:99:66:4b:19: 64:64:46:19:7e:7e:7c:98:99:66:4b:19: 64:64:46:19:7e:7e:7c:98:99:66:4b:19: 64:64:159:7e:7e:7c:98:99:66:4b:19: 64:64:159:7e:7e:7c:98:99:66:4b:19: 64:64:159:7e:7e:7c:98:99:66:4b:19: 64:64:159:7e:7e:7c:98:99:66:4b:19: 61:66 Exponent: 65:37 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:6f:fb:1b:10:51:9b:159:7b:143:72:fb:74:84:49:3c:160: 15:5f:6d:32:7d:1a:27:48:49:3c:160:162:33:80:7c:12:26:161: 55:19:40:66:15:32:36:19:44:93:31:7e:38:36:79:72:26: 55:19:40:66:15:32:36:19:44:49:32:160:162:33:80:7c:12:26:161: 55:19:40:66:15:32:36:19:44:35:36:19:44:35:19:44:85:49: 65:19:40:66:15:32:74:18:27:48:49:32:10:46:16:33:80:7c:12:20:40: 55:19:40:66:15:37:28:15:40:46:15:37:28:45:40:42:43:40:47:10:24:40:42:43:40:47:42:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:47:40:40:40:47:47:40:40:47:40:40:47:40:40:47:47:40:40</pre>	Not before Aug 15 66/07/22 2013 GMT	
Subject Public Key Info: Public Key Algorithm: rsaEncryption Public-Key: (2048 bit) Modulus: 00:f4:36:84:25:27:20:bd:45:ff:6f:a2:fb:92:4f: 19:dc:22:b6:42:b6:42:b4:e6:35:8b:24:69:9d:a7: 13:2d:41:4e:cb:e4:59:a1:42:1f:03:ab:cc:32:53: c9:26:39:2d:94:b5:1f:c3:44:11:69:03:9a:35:eb: 34:6d:77:D2:2a:8b:ad:c6:fa:2a:0b:56:34:d7:77: 51:36:b9:df:ab:79:67:a8:ea:96:23:27:89:ec:ca: ac:1f:47:6b:a2:42:164:e3:5b:8b:44:44:e0:37:f7: 7b:53:68:da:53:fa:e0:eb:15:16:c3:44:df:e0:37:f7: 7b:53:68:da:53:fa:e0:eb:15:16:c3:44:df:21: 4f:a9:a4:57:66:cc:09:e1:38:15:41:44:e0:37:f7: 7b:53:68:da:53:fa:e0:eb:15:18:44:df:fa:21: 4f:a9:a4:57:66:cc:09:e1:38:14:35:fe:e6: be:5c:18:44:35:90:66:dd:90:81:53:55:c6:11:19: 64:64:44:19:7e:7e:7c:98:98:13:35:ce:1b:12: 4f:a9:a4:67:06:cc:09:18:13:35:ce:1b:12: 64:64:44:19:7e:7e:7c:98:98:13:35:ce:1b:12: 16:6d Exponent: 65:37 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:ef:fb:1b:08:35:94:43:cd:c6:e8:03:5f:11: 1b:96:d5:27:14:12:74:84:36:15:67:16:12:12: 4f:a9:a4:42:55:67:10:12:26:de: 53:94:10:db:23:a8:10:44:14:a5:16:11:12: 65:16:14:44:49:77:27:79:31:cc:d4:ca:55:97:78:45: 16:6d Exponent: 65:37 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:ef:fb:1b:08:35:94:43:40:c6:16:19:31:19:c7:26:c6: 53:94:10:db:23:a8:10:49:32:10:c2:66:c1:39:19:c3:38:10:c1:26:10:19: 65:16:17:12:72:49:32:45:15:77:14:27:48:44:39:19:c7:26:12: 15:16:16:15:14:15:14:12:14:44:13:19:c8:aa: ad:df:ae:5d:0e:ce:88:55:15:14:13:19:c8:aa: ad:df:ae:5d:0e:ce:88:55:15:12:14:88:41: 65:16:1f:0e:d5:15:14:13:13:12:14:88:15:12:15:14:18:13:19:c8:aa: ad:df:ae:5d:0e:ce:88:55:12:13:18:13:12:28:11:44:18:18:13:12:28:11:44:18:18:13:12:28:11:45:18:18:13:12:14:18:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:13:19:c8:1	Subject: $C = IIS_{ST} = California = L = Milnitas_O = Infineon_OU = DSS_CN = www.infineon_com$	
Public Key Algorith:: rsaEncryption Public Key (2048 bit) Modulus: 00:f4:36:84:25:27:20:bd:45:ff:6f:a2:fb:92:4f: 19:dc:02:b6:42:b6:e9:a1:14:ec:68:c7:6d:39:13: 97:8b:b1:6f:c2:94:cb:02:a0:53:8b:2d:90:de:a7: 11:2d:41:4e:cb:e1:45:9a:11:42:1f:03:a1:cc:32:53: c9:26:30:2d:9d:b5:1f:c3:4a:11:69:03:9a:35:eb: 34:6d:27:b2:2a:8b:ad:c6:fa:2a:0b:5e:34:d7:77: 51:36:b9:df:ab:79:c7:a8:ea:96:23:27:89:0e:ca: ac:1f:47:fb:a2:d2:d4:e7:72:2f:35:2e:1d:e5:4f: b6:9f:18:55:4a:a1:a5:dd:ff:a6:29:0e:d6: ac:be:90:bb:3b:ab:85:93:95:18:41:4c:00:37:f7: 7b:53:68:da:35:fa:e0:eb:15:88:ff:a5:fe:e6: be:5c:d8:4a:95:09:6e:4d:90:81:53:5c:e6:b1:19: 64:64:04:f9:7e:7e:7c:98:9e:30:44:19:12:99:e6:b6: b7:22:08:24:55:90:6e:4d:90:81:53:5c:e6:b1:19: 64:e4:04:f9:7e:7e:7c:98:9e:39:44:19:10:40: f3:bb:a7:27:79:31:cc:d4:ca:5a:69:7a:8e:76:9c: 16:cd Exponent: e5537 (0x10001) Signature Algorithm: sha256MitRSAEncryption 62:cf:5f:ef:fb:b0:a5:9b:fb:48:72:fe:4a:64:a7:91:7d:76: b6:18:da:b7:44:93:ab:18:51:53:65:26:11: 1b:06:ed:52:7d:1a:27:44:93:ab:16:55:26:16: 53:94:06:db:23:a8:19:40:44:ef:78:a8:19:76:22:66: 53:94:06:db:23:a8:19:40:44:ef:78:a8:19:78:22:66: 53:94:06:db:23:a8:19:40:44:ef:78:a8:19:78:22:26:de: 53:94:06:db:23:a8:19:40:44:65:78:43:38:07:c2:26:de: 53:94:06:db:23:a8:19:40:44:65:78:43:38:07:c2:26:de: 53:94:06:db:23:a8:19:40:44:65:78:43:38:07:c2:26:de: 53:94:06:db:23:a8:19:40:44:65:78:43:38:07:c2:26:de: 53:94:06:db:23:a8:19:40:44:65:78:43:38:07:c2:26:de: 53:94:06:db:23:a8:19:40:44:65:78:43:38:07:c2:26:de: 53:94:06:db:23:a8:19:40:44:65:78:43:38:07:c6:22:26:de: 53:94:06:db:23:a8:19:40:44:65:78:43:38:07:c6:22:26:de: 53:94:06:db:23:a8:19:40:44:65:78:43:38:07:65:26:15:65:26:15:78: d5:67:67:67:69:63:67:09:28:43:58:77:78:48:41:49:32:08:78:78:71:45:78:64:41:49:32:08:78:84:41:49:32:08:78:84:41:49:32:08:78:84:41:49:32:08:78:84:41:49:32:08:78:84:41:49:32:08:78:84:41:49:32:08:78:84:41:49:32:08:78:84:41:49:32:08:78:84:41:49:32:08:78:84:41:49:32:08:78:84:41:49:32:08:78:84:41:49:32:08:78:45:78:45:41:49:32:49:28:84:41:49:32:49:28:84:41:49:32:49:28	Subject Public Key Info:	
Public-Key: (2048 bit) Modulus: 00:f4:36:84:25:27:20:bd:45:ff:6f:a2:fb:92:4f: 19:dc:02:b6:42:b6:49:a1:14:ec:68:c7:6d:39:13: 97:8b:b1:6f:c2:94:cb:0c:a0:53:8b:24:69:d0:a7: 13:2d:41:4e:cb:e4:59:a1:42:1f:03:ab:cc:32:53: c9:26:30:2d:9d:b5:1f:c3:4a:11:69:03:9a:35:eb: 34:6d:27:b2:2a:8b:ad:c6:fa:2a:0b:5e:34:d7:77: 51:36:b9:df:ab:70:22:24:2b:cd:25:22:2d:e5:4f: b6:9f:18:58:4a:a1:a5:dd:ff:a6:29:0a:d7:e0:06: ac:be:96:bb:3b:ab:85:99:39:5b:84:14:c0:03:7f7: 7b:53:68:da:53:fa:e0:eb:eb:15:e8:fa:25:f0:e6: be:5c:d8:4a:5b:01:64:66:f0:44:27:90:e6:bb: b7:e2:0a:24:55:90:66:4d:9b:81:53:5c:e6:b1:19: 64:64:04:f0:r7:r7:99:31:ac:d4:c2:99:06:bb: b7:e2:0a:24:55:90:66:4d:9b:81:53:5c:e6:b1:10: 64:64:04:f0:r7:r7:99:31:ac:d4:52:99:87:04: f3:bb:a7:27:79:31:cc:d4:c5:80:7a:82:76:9c: 16:cd Exponent: 65:57 (0x10001) Signature Algorithm: sha2560H18:31:f6:23:01:7d:76: b8:18:da:73:a8:99:40:44:e6:67:94:29:177:75: b5:18:da:73:a8:99:40:44:e6:67:94:29:177:75: b5:18:da:73:a8:99:40:44:e6:67:94:29:177:75: b5:18:da:73:a8:99:40:44:e6:67:94:29:34:41:90:37:04: f3:cb:a7:27:79:31:cc:d4:c5:80:7a:82:76:9c: 15:6d Exponent: 65:57 (0x10001) Signature Algorithm: sha2560H18:31:7c:e3:36:17:07:75: b8:18:da:77:44:3e:31:80:d0:72:38:95:26:11:8b:d0:d2: f3:e4:09:e2:38:99:40:44:e2:66:72:95:26:12: b:6f:e3:fa:9d:09:e2:38:19:40:44:e2:33:80:77:10:22:6d:e1: 53:94:06:db:23:38:09:40:44:e2:33:80:77:10:22:6d:e1: 53:94:06:db:23:38:09:40:44:e2:33:80:77:10:22:6d:e1: 53:94:06:db:23:38:09:40:44:e2:33:80:77:10:22:6d:e1: 53:94:06:db:23:38:09:40:44:e2:33:80:75:10:22:6d:e1: 53:94:06:db:23:38:09:40:44:e2:33:26:75:10:22:6d:e1: 53:94:06:db:23:38:09:40:44:e2:33:26:75:10:22:6d:e1: 53:94:06:db:23:38:09:40:44:e2:33:26:25:10:55:26:11:8b:0b:da: f5:ec:19:92:ee:3f:41:37:24:82:31:46:83:16:26:31:46:88:77: 71:34:6e:d0:17:db:28:8f:11:ee:75:81:23:27:12:20:71: 9b:10:dc:fd:69:10:12:20:86:45:77:22:20:71: 9b:10:dc:fd:69:10:12:20:86:45:77:22:20:71: 9b:10:dc:fd:69:10:12:20:86:45:77:22:20:71: 9b:10:dc:fd:69:10:12:20:30:64:45:78:12:20:71: 9b:10:dc:fd:68:10:10:21:20:27:31:20:37	Public Key Algorithm: rsaEncryption	
Modulus: 00:f4:36:84:25:27:20:bd:45:ff:6f:a2:fb:92:4f: 19:dc:20:b6:42:b4:09:a1:14:ec:68:c7:6d:39:13: 97:8b:b1:6f:c2:94:cb:0c:a0:53:8b:2d:69:00:a7: 13:2d:41:4e:cb:e4:59:a1:42:1f:03:ab:cc:32:53: c9:26:30:2d:9d:b5:1f:c3:4a:11:69:03:9a:35:eb: 34:6d:27:b2:2a:8b:ad:c6:fa:2a:0b:5e:34:d7:77: 51:36:b9:df:ab:79:67:a8:ea:96:23:27:89:8e:ca: ac:1f:47:6b:a2:42:d4:e7:72:2f:35:2e:1d:e5:4f: b6:9f:18:58:4a:a1:a5:dd:ff:a6:29:0a:d7:e0:06: ac:be:99:bb:3b:ab:59:39:55:88:41:4c:00:37:f7: 7b:53:68:da:55:1f:e0:c6:49:06:15:e3:4d:77:72: 51:36:b9:df:55:1f:e0:c6:47:06:15:e3:4d:77:72: 7b:53:68:da:55:1f:e0:c6:49:06:15:e3:44:d7:21: 4f:a9:a4:57:66:cc:d9:d6:55:e3:4d:d7:29:90:e6:b6: b7:e2:0a:24:55:90:6e:4d:9b:13:35:ce:6b:119: 64:64:04:f9:7e:7e:7c:98:9e:3a:46:90:46:12:104: f3:bb:a7:27:79:31:cc:d4:ca:5a:69:7a:8e:76:9c: 16:6d Exponent: 65537 (0x10001) Signature Algorithm: sha2560:thRSAEncryption 62:cf:5f:ef:fb:b0:fb:18:13:17e:ea:3d:65:c6:3:35:11: 1b:66:ed:52:7d:1a:27:48:49:3c:1b:62:33:80:7c:1b:22:6c:de: 53:94:06:cb:23:a8:19:44:49:3c:1b:62:33:80:7c:1b:22:6c:de: 53:94:06:cb:23:a8:19:44:49:3c:1b:62:33:80:7c:1b:22:6c:de: 53:94:06:cb:23:a8:19:44:49:3c:1b:62:33:80:7c:1b:22:6c:de: 53:94:06:cb:23:a8:19:44:49:3c:1b:62:33:80:7c:1b:22:6c:de: 53:94:06:cb:23:a8:19:44:49:3c:1b:62:33:80:7c:1b:22:6c:de: 53:94:06:cb:23:a8:19:44:49:3c:1b:62:33:80:7c:1b:22:6c:de: 53:94:06:cb:23:a8:19:44:49:3c:1b:62:33:80:7c:1b:22:6c:de: 53:94:06:cb:23:a8:19:44:49:3c:1b:62:33:80:7c:1b:22:6c:de: 53:94:06:cb:23:a8:19:44:49:3c:1b:62:33:80:7c:1b:22:6c:de: 53:94:06:cb:23:a8:19:44:49:3c:1b:62:33:80:7c:1b:22:6c:de: 53:94:06:cb:23:a8:19:44:49:3c:1b:62:33:80:7c:1b:22:6c:de: 53:94:06:cb:23:a8:19:44:49:3c:1b:62:33:80:7c:1b:22:6c:de: 53:94:06:cb:23:a8:19:44:49:3c:1b:62:45:76:12:20:fd: 61:87:6d:ff:02:10:00:21:84:60:37:23:80:55:20:fd:1b:7b:7b: d0:ff:04:ff:02:10:02:88:f1:ea:87:17d:75: 61:87:6d:ff:02:10:02:88:f1:ea:87:17d:75: d0:ff:04:ff:02:10:02:88:f1:ea:87:17d:75: d0:ff:04:ff:07:99:80:44:43:79:03:63:27:00:20:75: d0:cb:7c:79	Public-Key: (2048 bit)	
<pre>00:f4:36:84:25:27:20:bd:45:ff:6f:a2:fb:92:4f: 19:dc:02:b6:42:b6:e9:a1:14:ec:68:c7:b9:24:f; 97:8b:b1:6f:c2:94:cb:0e:a0:33:8b:2d:09:d0:a7: 13:2d:41:40:cb:e4:59:a1:42:1f:03:ab:cc:32:53: c9:26:30:2d!9d:b5:1f:c2:44:11:60:39:9a:35:eb: 34:6d:27:b2:2a:8b:ad:c6:fa:2a:0b:5e:34:d7:77: 51:36:b9:df:ab:79:67:a8:ea:96:23:27:89:0e:ca: ac:11:47:6b:a2:d2:d4:e7:72:2f:35:2e:1d:e5:4f: b6:9f:18:58:4a:a1:a5:dd:ff:a6:29:0a:d7:e0:06: ac:b9:0b:b1:ab:85:95:95:58:34:4c:00:37:f7: 7b:53:68:da:53:fa:e0:eb:eb:15:e8:fa:a5:fe:e6: be:c5:d8:4a:51:fa:e0:eb:eb:15:e8:fa:a5:fe:e6: bf:c5:d8:4a:51:fa:0e:eb:eb:15:e8:fa:a5:fe:e6: bf:c3:d8:45:76:16:20:99:108:15:35:ca:e6:11:19: c64:64:04:f9:7e:7e:7c:29:99:e3:a4:45:39:70:44: df:3c:10:ec:6e:1f':93:e3:24:d4:52:39:70:44: df:3c:10:ec:6e:1f':93:e3:24:d4:52:39:70:44: df:3c:10:ec:6e:1f':93:e3:39:70:44:20:35:d1: b1:67:27:79:31:cc:d4:ca:53:70:20:11:10:10:10:10:10:10:10:10:10:10:10:10</pre>	Modulus:	
<pre>19:dc:02:b6:42:b4:e9:a1:14:ec:68:C7:6d:39:13: 97:8b:b1:6f:c2:94:cb:0c:a0:53:8b:2d:69:d0:a7: 13:2d:41:4e:cb:e4:59:a1:42:1f:03:ab:cc:32:53: c9:26:30:2d:9d:b5:1f:c3:4a:11:69:83:9a:35:eb: 34:6d:27:b2:2a:8b:ad:c6:fa:2a:0b:5e:34:d7:77: 51:36:b9:df:ab:79:67:a8:ea:96:23:27:89:0e:ca: ac:1f:47:6b:a2:d2:d4:e7:77:22f:35:2e:1d:e5:4f: b6:97:18:58:4a:a1:a1:3d:df:ff:a6:29:0a:d7:e0:06: ac:be:90:bb:3b:ab:85:93:95:b8:41:42:06:37:f7: 7b:53:68:td:35:4a:e0:eb:15:1e:8f:a2:6b:15:e4:e6: be:5c:d8:4a:55:90:ed:16:4:66:67:04:27:4d:a4:df:21: 4f:a9:a4:57:66:cc:d9:d6:56:30:f7:29:09:e6:b6: b7:2e:32:45:59:06:e4:d9:58:15:55:ce:61:19: 64:64:04:f9:7e:7c:79:89:9e:3a:a4:69:46:b1:10: 64:64:04:f9:7e:77:93:1cc:d4:c5:36:67:a8:e76:9c: 16:6d Exponent: 65:57 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:57:ef:f5:f6:b0:23:a8:09:fb:04:83:17e:a8:3d:6c:5e:03:35:d1: 1b:96:ed:52:7d:1a:27:48:49:3c:b1:26:cd: 55:37:49:44:ed:65:f7:b0:23:48:00:26:cd:97:6c: b5:0f:e2:3a:8b:94:4e:c6:5f:7b:128:46:c3:97:6c: b5:0f:e2:3a:8b:94:4e:c6:5f:7b:128:46:c3:80:77:6c: b5:0f:e2:3a:8b:94:4e:c6:5f:7b:128:45:19:61:46: 55:37:44:3f:1d:e5:a1:19:61:24:23:38:87:62: d5:0f:e2:3a:8b:94:4e:c6:5f:7b:18:34:39:61:82:44: 07:65:fb:e7:0b:ed:5f:37:8e:35:19:4f:18:24:44: 07:65:fb:e7:0b:ed:5f:37:8e:35:19:4f:18:24:44: 07:65:fb:e7:0b:ed:5f:37:8e:45:19:19:61:44:24:20:71: 95:b4:f:a1:41:a1:14:21:31:80:18:31:72:a8:31:66:31:96:94: 82:c4:b1:a1:15:14f:8b:16:65:72:a2:85:22:07:12: 95:b4:61:45:17:45:14:45:12:34:8b:94:44:55:57:8e:45:19:44:87: 71:34:6e:d0:17:d0:22:84:19:31:38:05:15:22:07:12: 95:b4:6f:45:17:d0:24:19:23:38:19:44:13:18:83:17:14:19:c8:a3: 0f:18:16:d1:51:74:12:14:18:12:14:18:10:12:14:18:18:87: 71:34:6e:d0:17:d0:22:83:19:94:13:37:94:6f:15:22:07:11: 95:b4:6f:49:16:40:17:d0:22:83:19:44:85:15:22:07:11: 95:b4:6f:49:16:40:17:d0:22:83:45:17:45:12:20:71: 95:b4:6f:40:17:d0:22:83:45:17:45:12:20:71: 95:b4:6f:40:17:d0:22:83:45:17:45:12:20:71: 95:b4:6f:40:17:d0:22:83:45:17:45:12:20:71: 95:b4:6f:40:17:d0:22:83:45:17:45:12:20:71: 95:b4:6f:40:45:17:d0:22:83:45:17:45:12:20:71: 95:b4:6f:40:45:17:d0:22:83:4</pre>	00:f4:36:84:25:27:20:bd:45:ff:6f:a2:fb:92:4f:	
97:8b:10:6f:c2:94:cb:0c:a9:3:8b:2d:69:d0:a7: 13:2d:41:4e:cb:e4:59:a1:42:1f:03:ab:cc:32:53: c9:26:30:2d:9d:b5:1f:c3:4a:11:69:03:9a:35:eb: 34:6d:27:b2:2a:8b:ad:c6:fa:2a:70:55:eb:ca: ac:1f:47:6b:a2:d2:d4:e7:72:2f:35:2a:1d:e5:4f: b6:9f:18:58:4a:a1:a5:d0:ff:a2:00:37:f7: 7b:53:68:da:53:fa:e0:eb:eb:15:e8:fa:a5:fe:e6: be:c:d8:t4:3b:01:64:f6:67:04:27:4d:a4:f1:21: 4f:a9:a4:55:90:6e:d4:9b:15:s5:ce:61:04: c64:64:04:f9:7e:7e:7c:98:9e:3a:a4:69:46:b1:da: df:3c:10:ec:6e:1f:93:61:f2:20:04:52:39:87:04: f53:bb:a7:77:97:31:cc:d4:ca:5a:69:7a:8e:76:9c: 16:6d Exponent: 65:37 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:ef:fb:b6:18:18:18:18:16:c5:26:17:29:98:7b:04: 53:94:06:d0:23:a8:b9:16:14:22:26:d4:28:29:87:d1: 1b:96:ed:52:7d:1a:27:48:49:1cb:28:35:d1: 1b:96:ed:52:7d:1a:27:48:49:1cb:28:35:d1: 1b:96:ed:52:7d:1a:27:48:49:1cb:28:35:d1: 1b:96:ed:52:7d:1a:27:48:49:1cb:28:35:d1: 1b:96:ed:52:7d:1a:27:48:49:1cb:28:18:13:7c:65:16: 53:94:06:d0:23:a8:b9:16:14:37:26:26:16:18:18:18:17:65:26:16: 53:94:06:d0:23:a8:b9:16:14:37:26:26:de: 53:94:06:d0:23:a8:b9:16:14:37:26:26:de: 53:94:06:d0:23:a8:b9:16:14:37:2a:05:36:52:61:19:16:14:12:16:14:14:14:14:14:14:14:14:14:14:14:14:14:	19:dc:02:b6:42:b4:e9:a1:14:ec:68:c7:6d:39:13:	
13:2d1:41:4e:cb:e4:59:a1:42:1f:03:ab:cc:32:53: c9:26:30:2d:9d:b5:1f:c3:4a:11:69:39:35:eb: 34:6d:27:b2:2a:8b:ad:c6:fa:2a:0b:5e:34:d7:77: 51:36:b9:df:ab:79:67:a8:ea:96:23:27:89:0e:ca: ac:1f:47:fb:a2:d2:d4:e7:72:2f:35:2e:1d:e5:4f: b6:9f:18:58:4a:a1:a5:dd:ff:a6:29:0a:d7:e0:06: ac:be:90:bb:3b:ab:85:93:95:b8:41:4c:00:37:f7: 70:53:68:td:35:1f:ae:0e:be:bb:15:e8:fa:a5:fe:e6: be:5c:d8:4a:55:1f:ae:0e:be:bb:15:e8:fa:a5:fe:e6: be:5c:d8:4a:55:99:6e:4d:9b:81:53:5c:e6:b1:19: 64:64:04:f9:7e:7e:7c:98:99:3a:a4:69:46:b1:10a: df:ac:10:ec:6e:1f:93:06:17:29:90:e6:b6: b7:e2:0a:24:55:99:6e:4d:9b:81:53:5c:e6:b1:19: 64:64:04:f9:7e:7e:7c:98:99:3a:a4:69:46:b1:10a: df:ac:10:ec:6e:1f:93:06:17:29:90:70:04 f3:bb:a7:27:79:31:cc:d4:ca:5a:69:7a:8e:76:9c: 16:6d Exponent: 65:537 (0x10001) Signature Algorithm: sha256MithRSAEncryption 62:cf:5f:ef:ef:bb:06:a5:9b:fb:48:72:fe:44:64:a7:91:7d:76: b8:18:1a:b7:a4:39:31:80:18:31:7e:a8:3d:6c:5e:03:35:d1: 1b:96:ed:52:7d:1a:27:48:49:3c:bd:62:33:80:72:62:de: 53:94:06:db:23:a8:09:40:4e:c6:61:f1:93:19:c6: df:ae:5d:0e:ce:89:56:2a:b5:55:78:ea5:19:4f:85:84: 07:65:fb:er:0b:0e:df:fi:37:2a:05:3b:65:2c:1f:8b:0b:da: f5:ec:19:a9:ee:3f:4a:15:d5:72:a0:35:5b:2e:96:eb: 61:87:6d:fb:27:2a:03:20:71:a3:19:78:a3:19:78:a3:10: 1b:96:ed:52:77:1a:27:79:31:27:48:49:32:37:d3:80:66:d3:96:94: 82:cb:b1:a1:51:4f:8b:1b:0b:c9:71:23:80:35:5b:22:06:eb: 61:87:6d:fb:27:84:0d:61:53:78:e5:51:94:48:87: 71:44:6e:d0:17:a0:22:38:a5:51:29:46:82:47:80:18:e1:31:47:80:18:14:78:23:80:18:14:78:23:87:45:40:16:37:22:82:15:14:18:14:18:14:78:23:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:14:18:11:14:18:14:18:14:18:14:18:14:18:14:18:18:14:11:	97:8b:b1:6f:c2:94:cb:0c:a0:53:8b:2d:69:d0:a7:	
<pre>c9:26:30:2d:9d:b5:1f:c3:4a:11:69:83:9a:35:eb: 34:6d:27:b2:2a:8b:ad:c6:fa:2a:0b:5e:34:d7:77: 51:36:b9:df:ab:79:67:a8:ea:96:23:27:89:0e:ca: ac:1f:47:6b:a2:d2:d4:e7:72:2f:35:2e:1d:e5:4f: b6:9f:18:88:4a:1a:5d:ef:fa:62:98:ad7:a0:06: ac:be:90:bb:3b:ab:85:93:95:b8:41:4c:00:37:f7: 7b:53:68:da:55:1f:ae:0e:be:b1:5e:ef:fa:a5:fe:e6: be:5c:d8:4a:5b:01:64:66:67:04:27:4d:a4:df:21: 4f:a9:a4:57:66:cc:d9:d6:5e:63:0f:29:90:e6:b6: b7:e2:0a:24:55:99:6e:4d:9b:81:53:5c:e6:b1:19: 64:64:04:179:77:7e:77:79:89:e9:3a:44:69:46:b1:da: df:3c:10:ec:6e:1f:93:61:f2:20:d4:52:39:87:04: f3:bb:a7:27:79:31:cc:d4:ca:5a:69:7a:8e:76:9c: 16:6d Exponent: 65537 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:ef:fb:b0:a5:9b:fb:48:72:fe:4a:64:a7:91:7d:76: b8:18:da:b7:a4:3e:31:80:18:31:7e:a8:3d:6c:5e:08:35:d1: 1b:56:d1:27:d1:a2:7d:1a:27:44:94:3c:16:28:40:ec:8e:79:6c: 53:94:06:db:23:a8:b9:40:4e:c6:67:b0:28:40:0e:8e:79:6c: b5:61:39:ee:3f:4a:3f:1d:e5:fa:f6:76:b1:a3:19:e6:aa: ad:df:ae:5d:0e:ce:89:55:25:55:55:8e:a5:19:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:a1:5:55:57:8e:a5:19:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:a1:5:37:6e:c5:20:42:80:04:82:40:ec:8e:79:6c: b5:0d:a1:51:4f:8b:b6:05:55:75:8e:a5:19:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:a1:5:19:e5:25:78:ea:51:9:46:48:87: 71:14:6e:d0:17:40:32:37:20:36:88:cf:38:19:64:82:40:ec:8e:79:6c: b5:0f:e7:0b:ed:ff:02:a1:50:55:57:8e:a5:19:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:a1:30:68:cf:38:19:64:48:87: 71:14:6e:d0:17:40:28:11:51:4f:8b:12:16:16:7:15:23:19:e6:123:19:e7:12:39:e0:41: 31:15:14f:8b:16b:16:57:2a:23:88:15:2e:19:6e: 61:87:6d:ff:7f:54:cf:36:30:27:2a:23:88:35:52:2e:96:eb: 61:87:6d:ff:7f:54:cf:36:36:27:2a:23:88:35:52:2e:96:eb: 61:87:6d:ff:7f:54:cf:36:36:35:72:99:380:41:37:89:53:37:9d:cf:57: db:ff:d3:1c:44:e3:dc:7a:d7:38:46:80:13:19:df:57: db:ff:d3:1c:44:e3:dc:7a:d7:38:46:80:13:19:df:57: db:ff:d3:1c:44:e3:dc:7a:d7:38:46:80:13:19:df:57: db:ff:d3:1c:44:e3:dc:7a:d7:38:46:80:13:19:df:57: db:ff:d3:1c:44:e3:dc:7a:d7:38:46:80:13:19:df:57: db:ff:d3:1c:44:e3:dc:7a:d7:38:46:80:13:19:df:57: db:ff:d3:1c:44:e3:dc:7a:d7:38:46:80:13:10:</pre>	13:2d:41:4e:cb:e4:59:a1:42:1f:03:ab:cc:32:53:	
34:6d:27:b2:2a:8b:ad:c6:fa:2a:0b:5e:34:d7:77: 51:36:b9:df:ab:79:67:a8:ea:96:23:27:89:0e:ca: ac:1f:47:6b:a2:d2:d4:e7:72:2f:35:2e:1d:e5:4f: b6:9f:18:58:4a:a1:a5:dd:ff:a6:29:0a:d7:e0:06: ac:be:90:bb:3b:ab:85:93:95:b8:41:4c:060:37:f7: 7b:53:68:da:53:fa:e0:eb:eb:15:e8:fa:a5:fe:e6: be:5c:d8:4a:5b:01:64:66:67:04:27:4d:a4:df:21: 4f:a9:a4:57:66:cc:09:66:26:30:97:29:90:e6:b6: b7:e2:0a:24:55:90:6e:4d:9b:81:53:5c:e6:b1:19: 64:64:04:f9:7e:7e:7c:98:9e:3a:a4:69:46:b1:da: df:3c:10:ec:6e:1f:93:61:f2:20:44:52:39:87:04: f3:bb:a7:27:79:31:cc:d4:ca:5a:69:7a:8e:76:9c: 16:6d Exponent: 65537 (0x10001) Signature Algorithms: haz56WithRSAEnryption 62:cf:5f:ef:bb:05:a5:9b:fb:48:72:fe:4a:64:a7:91:7d:76: b8:18:da:b7:a4:3e:31:80:18:31:7e:a8:3d:6c:5e:03:35:d1: 1b:96:ed:52:7d:1a:27:48:49:3c:bd:62:38:80:7c:b2:26:de: 53:94:06:db:23:a8:109:40:4e:c6:7b:10:28:46:0e:8e:79:6c: b5:0f:e3:fa:9d:0e:db:16:37:2a:e5:1b:55:26:1f:8b:0b:da: f5:cc:19:9i:ee:3f:44:3f:1d:05:15:27:8e:a5:19:1f:8b:0b:da: f5:cc:19:9i:ee:3f:44:3f:1d:05:15:57:8e:a5:19:1f:8b:80:da: f5:cc:19:9i:ee:3f:44:3f:1d:05:15:57:8e:a5:19:1f:8b:80:da: f5:cc:19:9i:ee:3f:44:3f:1d:05:15:57:8e:a5:19:4f:88:84: 07:05:fb:e7:0b:ed:ff:02:e1:a6:37:23:80:25:52:96:eb: 61:87:6d:f9:7f:55:c0:36:32:70:03:80:8c:f5:81:1d:8e:87: 71:34:6e:d0:17:db:12:88:ff:8e:1d:38:87: 71:34:6e:d0:17:db:12:88:ff:8e:1d:38:87: 71:34:6e:d0:17:db:12:88:ff:8e:1d:8e:87: 71:34:6e:d0:17:db:12:88:ff:8e:1d:8e:87: 71:34:6e:d0:17:db:12:88:ff:8e:1d:8e:87: 71:34:6e:d0:17:db:12:88:ff:8e:1d:8e:87: 71:34:6e:d0:17:db:12:88:ff:8e:1d:8e:87: 71:34:6e:d0:17:db:12:88:ff:8e:1d:8e:87: 71:34:6e:d0:17:db:12:88:ff:8e:1d:8e:87: 71:34:6e:d0:17:db:12:88:ff:8e:1d:8e:87: db:ff:d8:1c:4d:e2:dc:7a:d7:38:46:88:18:b0:21:16:7e:51: db:ff:d8:1c:4d:e2:dc:7a:d7:38:46:88:18:b0:21:16:7e:51: dd:c6:4:68:d3:b7:d9:e1:9e:7c:7a:a9:39:64:3f:b4:85:95: 6d:eb:7c:79	c9:26:30:2d:9d:b5:1f:c3:4a:11:69:03:9a:35:eb:	
<pre>51:36:b9:df:ab:79:67:a8:ae:96:23:27:89:0e:ca: ac:1f:47:6b:a2:d2:d4:e7:72:2f:35:2e:1d:e5:4f: b6:9f:18:58:4a:a1:a5:dd:ff:a6:29:0a:d7:e0:06: ac:be:90:bb:3b:ab:85:93:95:b8:41:4c:00:37:f7: 7b:53:68:tad:53:fa:e0:eb:eb:15:e8:fa:a5:fe:e6: be:5c:d8:4a:5b:01:64:66:67:04:27:4d:a4:df:21: 4f:a9:a4:57:66:cc:d9:d6:5e:63:0f:29:90:e6:b6: b7:22:0a:24:55:90:6e:4d:9b:81:53:5c:e6:b1:19: 64:64:04:f9:7e:7e:7c:98:9e:3a:a4:69:46:b1:da: df:3c:10:ec:6e:1f:93:61:f2:20:d4:52:39:87:04: f3:bb:a7:27:79:31:cc:d4:ca:5a:69:7a:8e:76:9c: 16:6d Exponent: 65537 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:ef:fb:b0:a5:9b:fb:48:72:fe:4a:64:a7:91:7d:76: b8:18:da:b7:a4:3e:31:80:18:31:7e:a8:3d:66:5e:03:76:12:26:de: 53:94:06:db:23:a8:b9:40:4e:c6:67:b0:28:46:0e: 28:77:6c: b5:0f:e3:fa:ed:db:12:a8:b9:e4:26:33:80:7c:b2:26:de: 53:94:06:db:23:a8:b9:40:4e:c6:67:b0:28:46:0e: 28:79:6c: b5:0f:e3:fa:ed:db:12:a3:b9:40:4e:c6:75:b0:a5:96:f4:a0:e3:e7:96:c1 b5:0f:e3:fa:ed:db:12:a3:b9:f4:a4:f1:d::5fa:f6:76:b1:a3:19:c6:aa: ad:df:ae:5d:0e:cc:89:56:2a:b5:c5:57:8e:a5:19:4f:85:84: 07:55:fb:e7:0b:ed:ff:02:e1:06:23:74:f8:86:46:e6:8e:79:6c: b5:0f:e3:fa:ed:db:12:a3:b9:62:a3:b5:2e:96:eb: 61:87:0d:19:7f:55:79:93:80:41:37:88:16d:8e:87: 71:34:6e:d0:17:db:28:f1:ee:79:ed:23:74:45:82:12:71:47:45:84: 07:55:fb:e7:0b:ed:df:f1:22:17:48:19:12:37:43:80:18:17: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:80:b3:37:9d:cf:57: db:ff:d8:12:d4:e3:dc:7a:d7:38:46:80:17:b0:28:17:40:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:80:b3:37:9d:cf:57: db:ff:d8:12:d4:e3:dc:7a:d7:38:46:80:18:00:21:16:7e:51: 0d:cf:46:48:d5:d5:7d:12:37:80:23:7d:11:67:75: 0d:cf:46:48:d5:d5:7d:12:37:80:23:7d:11:67:75: 0d:cf:46:81:d5:d5:7d:12:37:80:23:7d:11:67:75: 0d:cf:46:81:d5:d5:7d:12:37:80:23:7d:11:67:75: 0d:cf:46:81:d5:d5:7d:12:37:80:23:7d:11:67:75: 0d:cf:46:81:d5:d5:7d:32:90:44:48:18:00:21:16:75:51: 0d:cf:46:48:d5:d5:7d:32:90:44:48:48:18:00:21:16:75:51: 0d:cf:46:48:d5:d5:7d:73:37:80:45:75:51: 0d:cf:46:48:d5:d5:7d:73:37:80:45:75:51: 0d:cf:46:48:d5:d5:7d:73:37:80:45:75:51: 0d:cf:47:79</pre>	34:6d:27:b2:2a:8b:ad:c6:fa:2a:0b:5e:34:d7:77:	
ac:1f:47.6b:a2:d2:d4:e7:72:2f:35:2e:1d:e5:4f: b6:9f:18:58:4a:a1:a5:dd:ff:a6:29:0a:d7:e0:06: ac:be:90:bb:3b:ab:85:93:95:b8:41:4c:00:37:f7: 7b:53:68:da:53:fa:e0:eb:eb:15:e8:fa:a5:fe:e6: be:5c:d8:4a:5b:01:64:66:67:04:27:4d:a4:df:21: 4f:a9:a4:57:66:cc:d9:d6:5e:63:0f:29:90:e6:b6: b7:e2:0a:24:55:90:6e:4d:9b:81:53:5c:e6:b1:19: 64:64:04:f9:7e:7c:98:9e:3a:a4:69:46:b1:da: df:3c:10:ec:6e:1f:93:61:f2:20:d4:52:39:87:04: f3:bb:a7:27:79:31:cc:d4:ca:5a:69:7a:8e:76:9c: 16:6d Exponent: 65537 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:ef:fb:b0:a5:9b:fb:48:72:fe:4a:64:a7:91:7d:76: b8:18:da:b7:a4:39:180:18:31:7e:a8:3d:6c:5e:03:35:d1: 1b:96:ed:52:7d:1a:27:48:49:3c:b6:22:61:ff:Bb:0b:da: f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:76:b1:a3:19:c8:aa: ad:df:ae:5d:0e:ce:89:56:2a:55:26:1f:Bb:0b:da: df:3e:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:76:b1:a3:19:c8:aa: ad:df:ae:5d:0e:ce:89:56:2a:55:27:8e:a5:19:4f:85:48: df:ae:6d:0f:7:2a:23:8c:a5:55:2e:96:eb: 61:87:6d:f9:7f:54:e0:36:a2:70:a3:60:85:21:96:94: 82:db:b1:a1:51:4f:Bb:bd:b6:97:2a:23:8c:a5:55:2e:96:eb: 61:87:6d:f9:7f:54:e0:36:a2:70:a3:60:88:1d:8e:87: 71:34:6e:d0:17:db:2e:88:f1:ae:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:38:84:137:89:b3:37:9d:cf:57: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:80:18:b0:21:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:30:18:b0:21:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:30:18:b0:21:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:30:18:b0:21:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:30:18:b0:21:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:30:18:b0:21:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:30:18:b0:21:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:30:18:b0:21:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:39:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:39:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:39:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:39:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:39:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:39:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:39:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:39:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:39:16:75: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:35:55:25	51:36:b9:df:ab:79:67:a8:ea:96:23:27:89:0e:ca:	
b:9f:18:58:4a:a1:a5:dd:ff:a6:29:0a:d7:e0:66: ac:be:90:bb:3b:ab:85:93:95:b8:41:4C:00:37:f7: 7b:53:68:da:53:fa:e0:eb:b5:15:e8:fa:a5:fe:e6: be:5c:d8:4a:5b:01:64:66:67:04:27:4d:a4:df:21: 4f:a9:a4:57:66:cc:d9:d5:E6:30:ff:29:90:e6:b6: b7:e2:0a:24:55:90:6e:4d:9b:81:53:5c:e6:b1:19: 64:64:04:f9:7e:7e:7c:79:89:9e:3a:a4:69:46:b1:1da: df:3c:10e:cc:e6:1f:22:04:d4:52:39:87:04: f3:bb:a7:27:79:31:cc:d4:ca:5a:69:7a:8e:76:9c: 16:6d Exponent: 65537 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:ef:fb:b0:a5:9b:fb:48:72:fe:4a:64:a7:91:7d:76: b8:18:da:b7:a4:3e:31:80:18:31:7e:a8:3d:6c:5e:03:35:d1: 1b:96:ed:52:7d:1a:27:48:49:3c:bd:62:33:80:7c:b2:26:de: 53:94:06:db:16:337:a4:3e:31:96:18:31:9c:28:46:0c:8e:79:6c: b5:ff:e3:fa:9d:0e:db:16:37:a3:19:28:46:10:28:47:66: b5:ff:e3:fa:9d:0e:db:16:37:a3:19:28:46:39:6:94: ad:df:ae:5d:0c:ce:89:56:2a:b5:c5:f7:8e:a5:19:4f:88:48: 07:65:fb:e7:b0:ed:16:37:13:68:e6:13:19:c8:aa: ad:df:ae:5d:0c:ce:89:56:2a:b5:c5:78:e3:19:4f:88:48: 07:65:fb:e7:b0:ed:16:32:13:68:e6:39:6:94: 82:db:b1:a1:51:4f:8b:bd:66:97:2a:23:8:c5:52:29:66:eb: 61:87:6d:f9:7f:54:c0:36:a2:70:03:b8:8c:f5:81:10:8e:78: 71:34:6e:d0:17:db:2e:8f:f1:ea:e7:9b:c3:f5:81:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:38:04:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:4d:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:44:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:44:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:44:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:44:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:44:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:44:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:44:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:44:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:44:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:44:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:44:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:44:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:44:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:44:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:40:e3:dc:7a:07:38:46:80:18:b0:21:16:7e:51: dd:cf:40:e3:dc:7a:07:38:46:80	ac:1f:47:6b:a2:d2:d4:e7:72:2f:35:2e:1d:e5:4f:	
ac:be:90:bb:3b:ab:85:39:55:b8:41:4c:00:37:f7: 7b:53:68:da:53:fa:e0:eb:eb:15:e8:fa:a5:fe:e6: be:5c:d8:4a:5b:01:64:66:67:04:27:4d:a4:df:21: 4f:a9:a4:57:66:cc:d9:d6:5e:63:06f:29:90:e6:b6: b7:e2:0a:24:55:90:6e:4d:9b:81:53:5c:e6:b1:19: 64:64:04:f9:7e:7e:7c:98:9e:3a:a4:69:46:b1:1da: df:3c:10:ec:6e:1f:93:61:f2:20:d4:52:39:87:04: f3:bb:a7:27:79:31:cc:d4:ca:5a:69:7a:8e:76:9c: 16:6d Exponent: 65537 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:ef:fb:b0:a5:9b:fb:48:72:fe:4a:64:a7:91:7d:76: b8:18:da:b7:a4:3e:31:80:18:31:7e:a8:3d:6c:5e:03:35:d1: 1b:96:ed:52:7d1:a:27:48:49:3c:bd:62:33:80:7c:b2:26:de: 53:94:06:db:23:a8:10:ec:6f:fb:02:8e:79:6c: b5:0f:e3:fa:9d:0e:db:16:37:2a:05:3b:65:26:1f:8b:0b:da: f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:7b:131:19:c8:aa: ad:df:ae:5d:0c:ce:89:56:2a:15:55:78:a5:19:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:96:94: 82:db:1b:13:15:14f:8b:bd:66:79:2a:23:8e:d5:55:29:eb: 61:87:6d:f7:f5:4c:03:6a:27:03:80:4f:37:2a:95:ab:c6:57: 71:34:6e:d0:17:db:28:8e:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:38:04:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: dd:dc:64:68:d3:17:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	b6:9f:18:58:4a:a1:a5:dd:ff:a6:29:0a:d7:e0:06:	
7b:53:68:da:53:fa:e0:eb:eb:15:e8:fa:a5:fe:e6: be:5c:d8:4a:55:90:16:46:66:70:427:4d:a4:df:21: 4f:a9:a4:57:66:cc:d9:d6:5e:63:061:29:90:e6:b6: b7:e2:0a:24:55:90:6e:4d:9b:81:53:5c:e6:b1:19: 64:64:04:f9:7e:7e:7c:98:9e:3a:a4:69:46:b1:da: df:3c:10:ec:6e:1f:93:61:1f:21:20:4d:52:39:87:04: f3:bb:a7:27:79:31:cc:d4:ca:5a:69:7a:8e:76:9c: 16:6d Exponent: 65537 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:ef:fb:b0:a5:9b:fb:48:72:fe:4a:64:a7:91:7d:76: b8:18:da:b7:a4:3e:31:80:18:31:7e:a8:3d:6c:5e:03:35:d1: 1b:96:ed:52:7d:1a:27:48:49:3c:bd:62:33:80:7c:b2:26:de: 53:94:06:db:23:a8:b9:40:4e:c6:67:b0:28:46:0c:8e:79:6c: b5:0f:e3:fa:9d:0e:db:137:2a:05:10:48:35:48: f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:7b:19:a3:19:c8:aa: ad:df:ae:5d:0c:ce:89:56:2a:b5:c5:77:8e:a5:10:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:90:94: 82:db:b1:a1:51:4f:8b:bd:b6:97:2a:23:8b:f5:2e:96:eb: 61:87:6d:f9:7f:54:c0:36:a2:70:03:6b:88:cf:58:1d:8e:87: 71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	ac:be:90:bb:3b:ab:85:93:95:b8:41:4c:00:37:f7:	
be:5c:d8:4a:5b:01:64:66:67:04:27:4d:a4:f21: 4f:a9:a4:57:66:cc:d9:d6:5e:63:0f:29:90:e6:b6: b7:e2:0a:24:55:90:6e:4d:9b:81:53:5c:e6:b1:19: 64:64:04:f91?e7:7e:7c:7e:98:99:3a:a4:69:46:b1:da: df:3c:10:ec:6e:1f:93:61:f2:20:d4:52:39:87:04: f3:bb:a7:27:79:31:cc:d4:ca:5a:69:7a:8e:76:9c: 16:6d Exponent: 65537 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:ef:fb:b0:a5:9b:fb:48:72:fe:4a:64:a7:91:7d:76: b8:18:da:b7:a4:3e:31:80:18:31:7e:a8:3d:6c:5e:03:35:d1: 1b:96:ed:52:7d:1a:27:48:49:3c:bd:62:33:80:7c:b2:26:de: 53:94:06:db:23:a8:b9:40:4e:c6:67:bb:28:46:02:8e:79:6c: b5:06f:e3:fa:9d:0e:db:16:37:2a:05:3b:65:26:1f:8b:0b:da: f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:ef:fb:10:a3:19:c8:aa: ad:df:ae:5d:0c:ce:89:56:2a:b5:c5:78:ea:51:19:c4:88:ef:88:44: 07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:96:94: 82:db:b1:a1:51:4f:8b:bd:b5:97:2a:23:8c:a5:b5:2e:90:eb: 61:87:6d:19:7:fb:4:0:36:a2:71:08:a5:b1:44:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	7b:53:68:da:53:fa:e0:eb:eb:15:e8:fa:a5:fe:e6:	
<pre>4+:a9:a4:57:66:cc:d9:d6:5e:63:d9:29:90:6e:4d:90:81:53:5c:e6:b1:19: 64:64:04:f9:7e:7e:7c:98:9e:3a:a4:69:46:b1:da: df:3c:10:ec:6e:1f:93:61:f2:20:d4:52:39:87:04: f3:bb:a7:27:79:31:cc:d4:ca:5a:69:7a:8e:76:9c: 16:6d Exponent: 65537 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:ef:fb:b0:a5:9b:fb:48:72:fe:4a:64:a7:91:7d:76: b8:18:da:b7:a4:3e:31:80:18:31:7e:a8:3d:6c:5e:08:35:d1: 1b:96:ed:52:7d:1a:27:48:49:3c:bd:62:33:80:7c:b2:26:de: 53:94:06:db:23:a8:19:40:44:c6:67:b0:28:46:0c:88:79:6c: b5:0f:e3:fa:9d:0e:db:16:37:2a:05:3b:65:26:1f:8b:0b:da: f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:76:b1:a3:19:c8:aa: ad:df:ae:5d:0e:ce:89:56:2a:b5:c5:57:8e:a5:19:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:e1:0e:32:37:d3:68:e6:d3:96:94: 82:db:b1:a1:51:4f:8b:bd:b6:97:2a:23:8c:a5:b5:2e:96:eb: 61:87:6d:19:7f:54:c0:36:a2:77:08:61:51:24:96:eb: 61:87:6d:19:7f:54:c0:36:a2:77:08:36:19:24:71:220:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79</pre>	be:5c:d8:4a:5b:01:64:66:67:04:27:4d:a4:df:21:	
b/ie2i9ai24i55i90i6ei40i90i81i53i5cie6i51i39i 64:64:04:04:f9:7e:7c:98:92:3a:a4:69:46:10:4a: df:3c:10:ec:6e:1f:93:61:f2:20:d4:52:39:87:04: f3:bb:a7:27:79:31:cc:d4:ca:5a:69:7a:8e:76:9c: 16:6d Exponent: 65537 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:ef:fb:b0:a5:9b:fb:48:72:fe:4a:64:a7:91:7d:76: b8:18:da:b7:a4:3e:31:80:18:31:7e:a8:3d:ec:5e:03:35:d1: 10:96:ed:52:7d:1a:27:48:49:3c:bd:62:33:80:7c:b2:26:de: 53:94:06:db:23:a8:b9:40:4e:c6:67:b0:28:46:0c:8e:79:6c: b5:0f:e3:fa:9d:0e:db:16:37:2a:05:3b:65:26:1f:8b:0da: f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:76:b1:a3:19:c8:aa: ad:df:ae:5d:0c:ce:89:56:2a:b5:c5:78:ea:51:91:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:96:94: 82:db:b1:a1:51:4f:8b:bd:b6:97:2a:23:8c:a5:b5:2e:96:eb: 61:87:6d:f9:7f:54:c0:36:a2:79:03:6b:88:cf:58:1d:8e:87: 71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 90:bd:c7:d0:1e:bf:25:79:93:80:41:37:80:b3:37:9d:cf:57: db:ff:d8:1c:44:e5:dc:ra:07:38:46:88:1b:20:16:7e:57: db:ff:d8:1c:44:e5:d7:d7:38:46:88:1b:20:16:7e:57: db:ff:d8:1c:44:e5:d7:d7:38:46:88:1b:20:16:7e:57: db:ff:d8:1c:44:e5:d7:a0:73:84:68:81:bb:20:16:7e:57: db:ff:d8:1c:44:e5:d7:a0:73:84:68:81:bb:20:16:7e:57: db:ff:d8:1c:44:e5:d7:a0:73:84:68:81:bb:20:16:7e:57: db:ff:d8:1c:44:e5:d7:a0:73:84:68:81:bb:20:16:7e:57: db:ff:d8:1c:44:e5:d7:a0:73:84:68:81:bb:20:16:7e:57: db:ff:d8:1c:44:e5:d7:a0:72:79:93:64:3f:b4:85:95: 6d:eb:7c:79	4+:a9:a4:57:66:cc:d9:d6:5e:63:04:29:90:e6:b6:	
bit	b/:22:0a:24:55:90:6e:4d:9b:81:55:5C:66:b1:19:	
<pre>ditisticite:e:1::9::0::0::2::9::0::0::2::9::0::0::0::0::0::0::0::0::0::0::0::0:</pre>	64:64:64:79:7/e:7/e:7/e:98:99:33:34:59:46:01:03:	
16:60 Exponent: 65537 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:ef:fb:b0:a5:9b:fb:48:72:fe:4a:64:a7:91:7d:76: b8:18:da:b7:a4:3e:31:80:18:31:7e:a8:3d:6c:5e:03:35:d1: 1b:96:ed:52:7d:1a:27:48:49:3c:bd:62:33:80:7c:b2:26:de: 53:94:106:db:23:a8:09:40:4e:c6:67:b0:28:46:00:88:79:6c: b5:0f:e3:fa:9d:0e:db:16:37:2a:05:3b:65:26:1f:8b:0b:da: f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:76:b1:a3:19:c8:aa: ad:df:ae:5d:0e:ce:89:56:2a:2b5:c5:57:8e:a5:19:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:96:94: 82:db:b1:a1:51:4f:8b:bd:b6:97:2a:23:8c:a5:b5:2e:96:eb: 61:87:6d:f9:7f:54:c0:36:a2:70:03:6b:88:cf:58:1d:8e:87: 71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	ut:sc:D0:ec:D0:L1:95:D1:T2:20:04:52:59:87:04:	
Exponent: 65537 (0x10001) Signature Algorithm: sha256WithRSAEncryption 62:cf:5f:ef:fb:b0:a5:9b:fb:48:72:fe:44:64:a7:91:7d:76: b8:18:da:b7:a4:38:18:31:7e:a8:3d:6c:5e:03:35:d1: 1b:96:ed:52:7d:1a:27:48:49:3c:bd:62:33:80:7c:b2:26:de: 53:94:06:db:23:a8:b9:40:4e:c6:67:b0:28:46:0c:8e:79:6c: b5:0f:e3:fa:9d:0e:db:16:37:2a:05:3b:65:26:1f:8b:0b:da: f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:76:b1:a3:19:c8:aa: ad:df:ae:5d:0c:ce:89:56:2a:b5:c5:57:8e:a5:19:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:96:94: 82:db:b1:a1:51:4f:8b:bd:b6:97:2a:23:8c:a5:b5:2e:96:eb: 61:87:6d:f9:7f:54:c0:36:a2:70:03:6b:88:cf:58:1d:8e:87: 71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:44:e3:d7:d7:38:46:88:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	15:00:a7:27:79:51:00:04:0a:5a:09:7a:00:70:90:	
Signature Algorithm: 5325(WithRSAEncryption 62:cf:5f:ef:fb:b0:a5:9b:fb:48:72:fe:4a:64:a7:91:7d:76: b8:18:da:b7:a4:3e:31:80:18:31:7e:a8:3d:6c:5e:03:35:d1: 1b:96:ed:52:7d:1a:27:48:49'3c:bd:62:33:80'7c:b2:26:de: 53:94:06:db:23:a8:b9:40:4e:c6:67:b0:28:46:0c:8e:79:6c: b5:0f:e3:fa:9d:0e:db:16:37:2a:05:3b:65:26:1f:8b:0b:da: f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:76:b1:a3:19:c8:aa: ad:df:ae:5d:0c:ce:89:55:2a:b5:c5:57:8e:a5:19:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:96:94: 82:db:b1:a1:51:4f:8b:bd:b6:b6:72:a2:38:ca5:b5:2e:96:eb: 61:87:6d:f9:7f:54:c0:36:a2:70:03:6b:88:cf:58:1d:8e:87: 71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	Exponent: 65537 (0x10001)	
Signed a Light Time Singer Signed and Signed a Light Construction of the Light Constructing the Light Constructing the Light Construction of th	Signature (sho)555 (WithRSAF)	
b8:18:da:b7:a4:3e:31:80:18:31:7e:a8:3d:6c:5e:03:35:d1: 1b:96:ed:52:7d:1a:27:48:49:3c:bd:62:33:80:7c:b2:26:de: 53:94:06:db:23:a8:bb:40:4e:c6:67:bb:28:46:0c:8e:79:6c: b5:06:e3:fa:9d:0e:0b:16:37:2a:05:3b:65:26:1f:8b:0b:da: f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:76:b1:a3:19:c8:aa: ad:df:ae:5d:0c:ce:89:56:2a:b5:c5:57:8e:a5:19:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:96:94: 82:db:b1:a1:51:4f:8b:bd:b6:97:2a:23:8c:a5:b5:2e:96:eb: 61:87:6d:f9:7f:54:c0:36:a2:70:03:6b:88:cf:58:1d:8e:87: 71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:77: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	62:cf:Sf:af:lh:h0:a5:9h:h1:42:77:fe:4a:f4:a7:91:7d:76:	
1b:96:ed:52:7d:1a:27:48:49:3c:bd:62:33:80:7c:b2:26:de: 53:94:06:db:23:a8:b9:40:4e:c6:67:b0:28:46:0c:8e:79:6c: b5:0f:e3:fa:9d:0e:db:16:37:2a:05:3b:65:26:1f:8b:0b:da: f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:76:b1:a3:19:c8:aa: ad:df:ae:5d:0c:ce:89:56:2a:b5:c5:57:8e:a5:19:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:96:94: 82:db:b1:a1:51:4f:8b:bd:b6:97:2a:23:8c:a5:b5:2e:96:eb: 61:87:6d:f9:7f:54:c0:36:a2:70:03:6b:88:cf:158:1d:8e:87: 71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:88:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	b8:18:da:b7:a4:3e:31:80:18:31:7e:a8:3d:6c:5e:03:35:d1:	
53:94:06:db:23:a8:b9:40:4e:c6:67:b0:28:46:0c:8e:79:6c: b5:0f:e3:fa:9d:0e:db:16:37:2a:05:3b:65:26:1f:8b:0b:da: f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:76:b1:a3:19:c8:aa: ad:df:ae:5d:0c:ce:89:56:2a:b5:c5:57:8e:a5:19:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:96:94: 82:db:b1:a1:51:4f:8b:bd:b6:97:2a:23:8c:a5:b5:2e:96:eb: 61:87:6d:f9:7f:54:c0:36:a2:70:03:6b:88:cf:58:1d:8e:87: 71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	1b:96:ed:52:7d:1a:27:48:49:3c:bd:62:33:80:7c:b2:26:de:	
b5:0f:e3:fa:9d:0e:db:16:37:2a:05:3b:65:26:1f:8b:0b:da: f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:76:b1:a3:19:c8:aa: ad:df:ae:5d:0c:ce:89:56:2a:05:c5:57:78:ea:51:91:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:96:94: 82:db:b1:a1:51:4f:8b:bd:b6:97:2a:23:8c:a5:b5:2e:96:eb: 61:87:6d:f9:7f:54:c0:36:a2:70:03:6b:88:cf:58:1d:8e:87: 71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	53:94:06:db:23:a8:b9:40:4e:c6:67:b0:28:46:0c:8e:79:6c:	
f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:76:b1:a3:19:c8:aa: ad:df:ae:5d:0c:ce:89:56:2a:b5:c5:57:8e:a5:19:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:96:94: 82:db:b1:a1:51:4f:8b:bd:b6:97:2a:23:8c:a5:b5:2e:96:eb: 61:87:6d:f9:7f:54:c0:36:a2:70:03:6b:88:cf:58:1d:8e:87: 71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:4d:e3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	b5:0f:e3:fa:9d:0e:db:16:37:2a:05:3b:65:26:1f:8b:0b:da:	
ad:df:ae:5d:0c:ce:89:56:2a:b5:c5:57:8e:a5:19:4f:85:84: 07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:96:94: 82:db:b1:a1:51:4f:8b:bd:b6:97:2a:23:8c:a5:b5:2e:96:eb: 61:87:6d:f9:7f:54:c0:36:a2:70:03:6b:88:cf:58:1d:8e:87: 71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:44:e3:dc:7a:d7:38:46:88:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	f5:ec:19:a9:ee:3f:4a:3f:1d:e5:fa:f6:76:b1:a3:19:c8:aa:	
07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:96:94: 82:db:b1:a1:51:4f:80:bd:b6:97:2a:23:8c:a5:b5:2e:96:eb: 61:87:6d:f9:7f:54:c0:36:a2:70:03:6b:88:cf:58:1d:8e:87: 71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	ad:df:ae:5d:0c:ce:89:56:2a:b5:c5:57:8e:a5:19:4f:85:84:	
82:db:b1:a1:51:4f:8b:bd:b6:97:2a:23:8c:a5:b5:2e:96:eb: 61:87:6d:f9:7f:54:c0:36:a2:70:03:6b:88:cf:58:1d:8e:87: 71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:99:cf:57: db:ff:d8:1c:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	07:65:fb:e7:0b:ed:ff:02:e1:0a:32:37:d3:68:e6:d3:96:94:	
61:87:6d:f9:7f:54:c0:36:a2:70:03:6b:88:cf:58:1d:88:87: 71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	82:db:b1:a1:51:4f:8b:bd:b6:97:2a:23:8c:a5:b5:2e:96:eb:	
71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71: 9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	61:87:6d:f9:7f:54:c0:36:a2:70:03:6b:88:cf:58:1d:8e:87:	
9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57: db:ff:d8:1c:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	71:34:6e:d0:17:db:2e:88:f1:ea:e7:9b:ca:45:7e:12:20:71:	
db:ff:d8:1c:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51: 0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	9b:bd:c7:d9:1e:bf:25:79:93:80:41:37:89:b3:37:9d:cf:57:	
0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95: 6d:eb:7c:79	db:ff:d8:lc:4d:e3:dc:7a:d7:38:46:80:18:b0:21:16:7e:51:	
6d:eb:/c:/9	0d:dc:64:68:d3:b7:d9:e1:9e:7c:7a:a9:93:64:3f:b4:85:95:	
	6d:eb:/c:/9	

Figure 18 Client/Endpoint Certificate

The just created certificate will be used as part of the TLS handshake process by the client to authenticate the server.

2.5.8 Creating Server Certificate

Now that we have created our Client/Endpoint key pair and certificate using SLx 9670 TPM2.0, we will replicate the process to create the needed certificate for the server.

Code Listing 19

```
001 tpm2tss-genkey -a rsa -s 2048 ./tpm2/tpm_wrapped_keys/server.key
    -p abc
002 openssl req -keyform engine -engine tpm2tss -config
    ./pki/openssl.cnf -key ./tpm2/tpm_wrapped_keys/server.key -new -out
    ./tpm2/csr/server.csr
003 openssl x509 -req -days 365 -in ./tpm2/csr/server.csr -CA
    ./pki/private/intCA.crt -CAkey ./pki/private/intCA.key -CAcreateserial
    -out ./tpm2/certs/server.crt
004 openssl x509 -in ./tpm2/certs/server.crt -noout -text
```

Reading Server Certificate

OPTIGA[™] TPM Application Note

Integration of TLS Functionality for OPTIGA[™] TPM SLx 9670 TPM 2.0

TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

<pre>pi@raspberrypi-os:~/tpm_hardened_tls \$ openssl x509 -in ./tpm2/certs/server.crt -noout -text</pre>
Certificate:
Data:
Version: 1 (0x0)
Serial Number:
ae:cb:a0:21:a1:9d:43:5a
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com
Validity
Not Before: Aug 16 06:21:23 2019 GMT
Not After : Aug 15 06:21:23 2020 GMT
Subject: C = US, ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:ba:45:99:4a:b5:23:b5:05:2d:01:c1:5a:37:e3:
17:9b:eb:4e:98:dc:36:63:bb:e9:9b:14:9f:76:88:
9d:15:25:8e:74:3e:b6:68:07:c1:e2:d8:31:55:82:
b5:c2:b5:43:c9:4e:f1:82:f4:fe:82:a7:30:f8:2f:
65:dc:0f:89:5a:87:58:35:69:4f:73:cb:89:46:98:
e2:25:de:05:75:63:31:61:e4:6e:5e:11:a6:a5:1b:
f9:46:7c:06:e3:95:f8:df:39:5a:6d:28:37:0c:94:
63:bf:c5:e3:1e:75:4f:9f:e2:a2:cb:3e:d0:bf:0c:
36:4f:22:60:20:90:62:2e:c9:d4:8e:4f:26:5f:5e:
a6:10:5a:f2:9b:9c:5a:a4:c0:25:d5:c0:88:cd:8f:
18:10:e1:32:9d:c3:d1:46:6f:41:c4:ef:be:f0:e9:
94:79:32:ae:51:27:9c:c4:69:5d:1e:4e:0c:fb:bb:
01:be:cc:72:5a:63:e5:53:3d:04:ef:02:e3:39:66:
07:be:87:4a:b9:17:5d:f7:ab:84:dc:d3:73:4b:06:
03:96:c5:e5:ff:a0:73:d6:04:4e:d1:9d:bb:d1:3e:
b0:3e:91:cd:77:82:27:7d:c0:9b:21:47:a7:0d:59:
18:ab:eb:1b:4d:f2:f7:eb:ef:40:bb:08:83:28:37:
10:1d
Exponent: 65537 (0x10001)
Signature Algorithm: sha256WithRSAEncryption
a+:97:ac:2b:20:d0:6e:+2:23:e0:c1:+b:+a:+1:bd:21:92:9c:
+8:40:d9:+a:33:38:90:/a:55:62:94:/e:69:8a:63:6C:8d:be:
84:00:34:14:74:49:26:88:dd:09:0217b:87:59:14:02
1a:69:ef:d3:e8:/b:3f:62:b/:c5:a2:65:1e:39:44:44:d1:c3:
50:23:53:62:62:62:60:60:11:60:96:63:08:48:63:75:13:88:07:
12:0/:ea:a5:51:94:20:cb:66:35:34:da:96:1d/:4:18:09:5b:
c3:tD:61:dt:dc:1D:5c:4c:ct:dc:34:DD:9D:5d:38:D4:98:c9:
47:16:07:53:00:41:68:40:33:76:C2:52:e4:D7:58:30:03:C7:
45:10:11:00:00:70:55:08:20:87:04:55:07:40:45:87:04:57:44:58:01:
50:a0:TE:00:02:31:01:/C:08:59:TT:C5:08:03:09:28:12:1/:
85:01:02:/0:08:48:95:82:04:04:0/17/108:6/196:51:14:56:
ca:cu:2u:00:00:90:1T:DD:Ce:94:4C:a1:a0:DU:SD:a9:81:7/2:
TC: 49: UQ: 44: ET: U2: T3: 44 / 20: 72: 30: 3D: EE: C5: T2: V0: 3C: GT:
Lu.Le./0:00.41:00:35:42:00:DC:45:28:54:94:21:00:
18:10:25:80

Figure 19 Server Certificate

Verifying the Certificate chain

Code Listing 20

```
001 openssl verify -CAfile ./pki/private/rootCA.crt -untrusted
./pki/private/intCA.crt ./tpm2/certs/server.crt
```

pi@raspberrypi-os:~/tpm_hardened_tls \$ openssl verify -CAfile ./pki/private/rootCA.crt -untrusted ./pki/private/intCA.crt ./tpm2/certs/server.crt ./tpm2/certs/server.crt: OK

Figure 20 Client Certificate Chain verification

2.6 Creating an OpenSSL S_Server

We will create now an OpenSSL server. For this purpose, we are using the local host capabilities to run this example on the same Linux machine.

Create an openssl S_Server instance using a terminal window session.





Code Listing 21 OpenSSL S_Server terminal window

```
001 openssl s_server -www -Verify 1 -cert ./tpm2/certs/server.crt -
key ./tpm2/tpm_wrapped_keys/server.key -keyform engin -engine tpm2tss -
accept 127.0.0.1:8444
```

Note: When ask for the password to use the key, input the set password during key creation "abc".

```
pi@raspberrypi-os:~/tpm_hardened_tls $ openssl s_server -www -Verify 1 -cert ./tpm2/certs/server.crt -key ./tpm2/tpm_wra
pped_keys/server.key -keyform engin -engine tpm2tss -accept 127.0.0.1:8444
verify depth is 1, must return a certificate
engine "tpm2tss" set.
Enter password for user key:
Using default temp DH parameters
ACCEPT
```

Figure 21 OpenSSL S_Server

2.7 Creating an OpenSSL S_Client

We will create an OpenSSL S_Client and connect through a TLS session with OpenSSL S_Server (The two terminal windows and services running on the same Linux machine).

Open a new terminal window and go to our root directory for this exercise (tpm_hardened_tls) and run the following command.

Create an OpenSSL S_Client.

Code Listing 22 OpenSSL S_Client terminal window

```
001 openssl s_client -cert ./tpm2/certs/client.crt -key
./tpm2/tpm_wrapped_keys/client.key -keyform engine -engine tpm2tss -
connect localhost:8444
```

Note:

Note that both S_Server and S_Client will be using TPM2-TSS Engine alongside SLx 9670 TPM2.0 to interact and establish a TLS session.

The output of the connection is divided in two parts

- a) The TLS handshake
- b) TLS Cipher

```
Note: When using self-signed certificates in OpenSSL, as a precaution it will give a warning stated as an "error": Verification error: self-signed certificate in certificate chain.
```

To verify that our certificate chain is valid we can run the following commands

As shown in Figure 22 and Figure 23 the complete TLS handshake process was successful, and the encrypted channel established.



TLS Hardening by OPTIGA™ SLx 9670 TPM2.0

pi@raspberrypi-os:~/tpm_hardened_tls \$ openssl s_client -cert ./tpm2/certs/client.crt -key ./tpm2/tpm_wrapped_keys/client.key -keyf orm engine -engine tpm2tss -connect localhost:8444
engine "tom2tss" set.
Enter password for user kev:
CONNECTED(00000004)
depth=0 C = US. ST = California, L = Milpitas, O = Infineon, OU = DSS, CN = www.infineon.com
verify error:num=18:self signed certificate
verify return 1
denthe C = US ST = California L = Milnitas O = Infineon OL = DSS (N = WWW infineon com
venify return 1
Centificate chain
Certificate chain
i./C-US/ST-California/L-Milpitas/O-Infineon/OU-DSS/CN-uww.infineon.com
1.7×10^{-10}
Server certificate
MIIDXJCCAKYCCQCuy6AhoZIDwJANBgKqhKIG9MøBAQsFADBXMQsWCQYDVQQGEWJV
UZE IMBEGAIDECAWKQ2FsaWZvcm5pYIERMA8GAIDEBWWIIWISCGI0YXMXEIAPBgNV
BAOMCE1u2m1u2W9uMQwwcgYDVQQLDANEU1MxGIAXBgNVBAMMEHd3dy5pbm2pbmVv
bi5jb20wHhcNMTkwODE2MDYyMTIzWhcNMjAwODE1MDYyMTIzWjBxMQswCQYDVQQG
EwJVUZETMBEGA1UECAwKQ2FsaWZvcm5pYTERMA8GA1UEBwwITW1scG10YXMxETAP
BgNVBAoMCE1uZm1uZW9uMQwwCgYDVQQLDANEU1MxGTAXBgNVBAMMEHd3dy5pbmZp
bmVvbi5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC6RZ1KtSO1
BS0BwVo34xeb606Y3DZju+mbFJ92iJ0VJY50PrZoB8Hi2DFVgrXCtUPJTvGC9P6C
pzD4L2XcD4lah1g1aU9zy4lGmOII3gV1YzFh5G5eEaalG/lGfAbjlfjf0VptKDcM
1G0/xeMedU+f4qLLPtC/DDZPImAgkGIuydSOTyZfXqYQWvKbnFqkwCXVwIjNjxgQ
4TKdw9FGb0HE777w6ZR5Mq5RJ5zEaV0eTgz7uwG+zHJaY+VTPQTvAuM5Zge+h0q5
F133q4Tc03NLBg0WxeX/oHPWBE7RnbvRPrA+kc13gid9wJshR6cNWRir6xtN8vfr
70C7CIMoNxAdAgMBAAEwDQYJKoZIhvcNAQELBQADggEBAK+XrCsg0G7yI+DB+/rx
vSGSnPhA2fozOJt6VWKUfmmKY2yNvoQANPR0SSbI3cnCe4e3h1kUwhpp79Poez9i
t8WiZR45RETRw1AiU9LSa9ARu5a1CEij9R00D/IH6aVf1CDLZiU02pYddPzJW8P7
Yd/cG1xMz9w0u5td0LSYyU8cB1Pb0WhLo37CUuS/WbyTz0MdEW7gfVNoLY/kw+/U
X6RY0VCm/ggCMWE8aEn/xdia2SgSE4XR3nDoSJWCBN0H97hn]]EIJNsrNK2hgkB+7
znRMoaawhamRcvvvn3llrv4vvHII X19g+268go83830doCh7hkI kØTEvElloV10h7Rgh
END_CERTIFICATE
subjects/C=LS/ST=California/L=Milnitas/O=Tofingon/OLEDSS/CN=WWW infingon com
Subject () costs of end to the first action of the second s
No client contificate CA names cont
No citeric Certificate CA names sent
CTIENT CERTIFICATE TYPES: NOA SIGN, DOA SIGN, ECDAA SIGN Deguesta General General Deguesta Deguesta Contra Contra Contra Contra Contra Contra Contra Contra Contra Contra Deguesta Contra Contr
<pre>Sharea kequested signature aigorithms: KSA+SHAS12:DSA+SHAS12:ECDSA+SHA384:DSA+SHA384:ECDSA+SHA384:RSA+SHA256:DSA+SHA256: CSCA+CCDSA+CCD2A+CCD2A+CSCA2A+CSCA2A+CSCA2A+SHA312:ECDSA+SHA384:DSA+SHA384:ECDSA+SHA384:RSA+SHA256:DSA+SHA256:</pre>
ECUSA+SHAZ30; KSA+SHAZ24; USA+SHAZ24; ECUSA+SHAZ24; KSA+SHA1; USA+SHA1; ECUSA+SHA1
Peer signing algest: SHA512
Server Lemp Key: X25549, 255 DITS

Figure 22 OpenSSL S_Client and S_Server TLS Handshake hardened by OPTIGA[™] SLx 9670 TPM2.0



TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

No client certificate CA names sent Client Certificate Types: RSA sign, DSA sign, ECDSA sign Requested Signature Algorithms: RSA+SHA512:DSA+SHA512:ECD HA256:RSA+SHA224:DSA+SHA224:ECDSA+SHA512:DSA+SHA1:DSA+SHA Shared Requested Signature Algorithms: RSA+SHA512:DSA+SHA ECDSA+SHA256:RSA+SHA224:DSA+SHA224:ECDSA+SHA224:RSA+SHA1: Peer signing digest: SHA512 Server Temp Key: X25519, 253 bits SSL handshake has read 2418 bytes and written 1419 bytes Verification error: self signed certificate	SA+SHA512:RSA+SHA384:DSA+SHA384:ECDSA+SHA384:RSA+SHA256:DSA+SHA256:ECDSA+S 1:ECDSA+SHA1 512:ECDSA+SHA512:RSA+SHA384:DSA+SHA384:ECDSA+SHA384:RSA+SHA256:DSA+SHA256: DSA+SHA1:ECDSA+SHA1
<pre>New, TLSv1.2, Cipher is ECDHE-RSA-AES256-GCM-SHA384 Server public key is 2048 bit Secure Renegotiation IS supported Compression: NONE Expansion: NONE No ALPN negotiated SSL-Session: Protocol : TLSv1.2 Cipher : ECDHE-RSA-AES256-GCM-SHA384 Session-ID: 4823FB3B980D967741E0542880122ABDBF8D9522B Session-ID-ctx: Here Free Content and the second second</pre>	F49002CF020AE05FAFB2AF8
PSK identity: None PSK identity hint: None SRP username: None TLS session ticket lifetime hint: 7200 (seconds) TLS session ticket: 0000 - 17 al 5e d5 f4 aa 9b 8a-4f 3a ac 85 4l 5l 99 6 0010 - ca f6 96 bb 25 55 ee f3-78 2c 1d 45 74 cf d7 5 0020 - 66 9f cd al cd fa 35 2l-0a 6b e3 4d a8 d3 c6 2 0030 - 92 c2 5f 1b b3 97 e3 cc-36 a5 2f 6a 1l 79 bb b 0040 - 89 9l 4b 9a 6a e4 el 88-14 04 93 7a 3f b0 d3 9 0050 - 3l 18 87 18 4a al 1b ad-fa bf ba 85 05 55 c6 5 0060 - 6f 3c 15 17 33 c8 6b a3-db 4l c8 2f 1a 34 82 2 0070 - ec 78 3e bf 5d 77 ff c5-13 37 da fe 88 ec 84 b 0080 - 63 5e 65 28 a8 39 13 a0-b1 2e d0 fe 6c 6a 60 c 0090 - 27 a5 5c 2e 4l 89 ca 9b-b4 fc 52 54 3c dc 7a 0 00a0 - 10 f0 8a c8 10 2e 8f ae-49 73 65 5l 66 dd 23 5 00b0 - b1 81 86 bd 27 4a 2c dd-cd 6f 08 bc c6 7l 96 00c0 - b9 1b f2 24 12 ab 27 5f-d7 ca ac 5d 10 8b fd 4 00d0 - d3 38 97 d7 6f 04 7l 06-35 54 7d 7l 3e 5c e6 1 00e0 - 5e 97 f6 7f 17 1a 32 af-be 9a fl 82 5l 1b dd 6 00f0 - ca d7 77 e4 0a dc 9c cd-12 3c 1c 74 1a 3d 5e f 0100 - 6e de a2 80 cb 16 f3 f7-7f 22 7d 72 02 98 5a 5 0110 - 9d 86 94 72 2l be 9f 9a-27 ff 37 2f 69 37 d3 3 0120 - cd 9l c6 de 57 55 a0 1b-d7 8a 3f 85 00 d6 8l 2 0130 - 38 56 16 7d e2 63 c7 a6-9c 92 ac 3a 69 ec b3 7 0140 - cc 90 ba fd 5f 0b 80 42-40 c7 48 ed 8c b5 65 0150 - 61 8b ab f4 fc 95 2a ac-2a df 74 74 29 c2 25 25 0150 - 61 8b ab f4 fc 95 2a ac-2a df 74 74 29 c2 25 27 0160 - f3 3 39 46 cf 2c 48 e0-4d 17 86 93 4f bd 60 0 0170 - ae fe 7a 77 9e 07 7c 3a-32 54 f9 0c 2e 7d 46 5 0180 - 18 06 2b d8 e2 ff 65 af-4b 4e eb el 23 07 a8 5 0190 - 0f 8f 5a 8e f4 c3 cd d0-50 56 11 ec 02 c6 0c d 01a0 - c9 f4 db e9 b 2f 8 d5 94-4d 08 9e e8 2a 9e 1c 0 01a0 - c9 f4 db e9 b 2f 8 d5 94-4d 08 9e e8 2a 9e 1c 0 01a0 - c9 f4 db e9 b 2f 8 d5 94-4d 08 9e e8 2a 9e 1c 0 01a0 - c9 f4 db e9 b 2f 8 d5 94-4d 08 9e e8 2a 9e 1c 0 01a0 - c9 f4 db e9 cf 8 f3 e7 6c-55 53 fb bd 56 11 d 01c0 - e0 b9 fa 1f b7 34 5a 3a-e3 bb 4b 5e 17 14 5f 4	<pre>e^0:AQ.n 4%U.x,Et.T f f5!.k.M/ 56./j.y. 6K.jz? b 1JU.[d o<3.k.A./.4 6 .x>.]w7 a c^e(.91j`. 1 '.\ART<.z. 3IseQf.#S a'J,.ooj 4\$'Q.e ew<t.t=^. e nYr.Z^ 2r!'.7/i7.2 5WU?% b 8V.J.ci{ fB@.H 8 a**.ttJ.R. c >9F.,H.MO.`. aYr d</t.t=^. </pre>
0100 - e0 FT 8a 57 51 85 84 91-29 7c 0c DT 05 42 65 D 01e0 - a2 a9 ca 9e 02 18 de 3e-14 d2 e8 84 fb f3 73 3 01f0 - 93 98 8e 70 a7 dd 52 94-1a 35 5c f6 34 27 18 c 0200 - 25 45 76 a6 10 c1 8a d4-c7 2d 4c 33 48 a9 d6 f	0>

Figure 23 OpenSSL S_Client and S_Server TLS Cipher

As a summary, this was the process that followed the TLS session.



TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

	lient		Server
1 Client Hello 5 Client Key Exchange Message	Supported Ciphers Random Number Session ID SMI	• Select Ciphers • Random Number • Session ID • SMI (empty)	Server Hello 2 Server Certificates 3 Server Hello Done 4
6 Key Generation	Premaster Key Generation		Key Generation 6
7 Cipher Spec. Exchange 8 Finish			7 Cipher Spec. Exchange
Hand-Shake Protocol	<		Finish 9
Record Protocol Application Data			Application Data

Figure 24 OpenSSL S_Client and S_Server TLS session flow

- **Client Hello**: The Client Hello is the first message in the TLS handshake from the client to the server. The Client Hello message includes the highest version of the TLS protocol the client supports, a random number generated by the client, cipher suites and the compression algorithm supported by the client, and an optional session identifier.
- Server Hello message is the response from the Server once it receives the Client Hello. The Server Hello is the first message from the server to the client. To be precise, the Server Hello is the first message from the server to the client, which is generated at the TLS layer. The Server Hello message includes the highest version of TLS protocol that both the client and the server can support, a random number generated by the server, the strongest cipher suite, and the compression algorithm that both the client and the server can support. The Server can use the TPM as source of entropy, that is, use TPM to generate the random number.
- **Server Certificates**: After the Server Hello message is sent to the client, the server sends its public certificate, along with other certificates, up to the root certificate authority (CA) in the certificate chain. The client must validate these certificates to accept the identity of the server.
- **Key Generation**: At this point, the client and the server have exchanged all the required materials to generate the master secret. The master secret is generated using the client random number, the server random number, and the premaster secret.
- **Change Cipher Spec:** Message to the server to indicates that all messages generated from here onward are protected with the keys already established.



TLS Hardening by OPTIGA[™] SLx 9670 TPM2.0

- Client Finish Message: Last message from the client to the server. It's the hash of the complete
 message flow in the TLS handshake encrypted by the already-established keys. Once the server
 receives the Finished message from the client, it responds back with the Change Cipher
 Spec Message. This indicates to the client that the server is ready to start communicating with the
 secret keys already established.
- **Server Finish Message**: This is like the Finished message generated by the client and includes the hash of the complete message flow in the handshake encrypted by the generated cryptographic keys. This completes the TLS handshake and here onward both the client and the server can send data over an encrypted channel.
- *Note:* When the server demands TLS mutual authentication, then the server will request the client certificate. The client certificate request message from the server includes a list of certificate authorities trusted by the server and the type of the certificate. After that, the server sends the Server Hello Done message to the client. This is an empty message that only indicates to the client that the server has completed its initial phase in the handshake.

If the server demands the client certificate, now the client sends its public certificate along with all other certificates in the chain up to the root certificate authority (CA) required to validate the client certificate.

• **Client Key Exchange Message**: After the Server Hello message is sent to the client, the server sends its public certificate, along with other certificates, up to the root certificate authority (CA) in the certificate chain. The client must validate these certificates to accept the identity of the server.

In the next section will use TShark as a sniffer to capture and log the TLS session. This tool will enable us to see the actual process that is followed as described in Figure 24.

Decoding SSL/TLS Traffic using TShark



3 Decoding SSL/TLS Traffic using TShark

TShark is a network protocol analyzer. It lets the capture of data packets from a live network or read packets from a previously save captured.

TShark native format of captured files is "pcapng", which is also the format used by WireShark[8].

We will be using TShark to decode the traffic between the S_Server and S_Client. This will allow to monitor and follow the TLS handshake exchange.

3.1 Installing TShark

Execute the following commands to install TShark on Raspberry Pi[®] 3B+/4 and verify that it installed correctly.

Code Listing 23

```
001 sudo apt-get update
002 sudo apt-get install tshark -y
003 tshark -v
```

```
pi@raspberrypi-os:~/tpm_sserver $ tshark -v
TShark (Wireshark) 2.6.7 (Git v2.6.7 packaged as 2.6.7-1~deb9u1)
```

Figure 25 TShark Install

3.2 Available Network Interfaces to use with TShark

Displaying available network interfaces that TShark can use

Code Listing 24

001 sudo tshark -D

```
pi@raspberrypi-os:~/tpm_sserver $ sudo tshark -D
Running as user "root" and group "root". This could be dangerous.
1. eth0
2. wlan0
3. any
4. lo (Loopback)
5. bluetooth0
6. nflog
7. nfqueue
8. usbmon1
9. ciscodump (Cisco remote capture)
10. randpkt (Random packet generator)
11. sshdump (SSH remote capture)
12. udpdump (UDP Listener remote capture)
```

Figure 26 Network Interface detected in Raspberry Pi[®] 3B+/4

Decoding SSL/TLS Traffic using TShark



3.3 Testing the Capture of Network Traffic with TShark

To test that all is working before moving on, we will capture a loopback transaction using the available Local Host: 127.0.0.1.

Open an additional terminal window (additional to the one we are using for TShark)

Code Listing 25 TShark terminal window

001	tshark -i 4		

Code Listing 26 Terminal window 2

001	ning 127 0 0 1	
001	ping 127.0.0.1	

To escape the capturing by TShark and the ping sequence use Linux terminal escape sequence "**ctrl + c**"

pi@ra	spberrypi-os:~/	<pre>tpm_sserver \$ tshark -i -</pre>		
Captu	ring on 'Loopba	ck'		
1	0.00000000	127.0.0.1 → 127.0.0.1	ICMP 98 Echo (ping) request id=0x04b2, seq=1/256, ttl=64	
2	0.000113953	127.0.0.1 → 127.0.0.1	ICMP 98 Echo (ping) reply id=0x04b2, seq=1/256, ttl=64 (request in 1)	
3	1.043414058	127.0.0.1 → 127.0.0.1	ICMP 98 Echo (ping) request id=0x04b2, seq=2/512, ttl=64	
4	1.043463379	127.0.0.1 → 127.0.0.1	ICMP 98 Echo (ping) reply id=0x04b2, seq=2/512, ttl=64 (request in 3)	
5	2.083403746	127.0.0.1 → 127.0.0.1	ICMP 98 Echo (ping) request id=0x04b2, seq=3/768, ttl=64	
6	2.083453379	127.0.0.1 → 127.0.0.1	ICMP 98 Echo (ping) reply id=0x04b2, seq=3/768, ttl=64 (request in 5)	
7	3.123400772	127.0.0.1 → 127.0.0.1	ICMP 98 Echo (ping) request id=0x04b2, seq=4/1024, ttl=64	
8	3.123447176	127.0.0.1 → 127.0.0.1	ICMP 98 Echo (ping) reply id=0x04b2, seq=4/1024, ttl=64 (request in 7)
9	4.163401855	127.0.0.1 → 127.0.0.1	ICMP 98 Echo (ping) request id=0x04b2, seq=5/1280, ttl=64	
10	4.163446227	127.0.0.1 → 127.0.0.1	ICMP 98 Echo (ping) reply id=0x04b2, seq=5/1280, ttl=64 (request in 9)
^C10	packets capture	d		



At the end of the capture TShark will indicate the number of packets that were captured. In this case 10.

```
pi@raspberrypi-os:~ $ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.194 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.122 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.113 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.112 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.102 ms
^C
--- 127.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4163ms
rtt min/avg/max/mdev = 0.102/0.128/0.194/0.035 ms
```

Figure 28 Terminal window 2

Decoding SSL/TLS Traffic using TShark



3.4 Capturing a TLS Session using TShark

Within the working directory (tpm_hardened_tls), create a new directory name for example "tls_log".

Additionally, we will open three terminal windows:

- 1. TShark terminal window
- 2. S_Server terminal window
- 3. S_Client terminal window

Code Listing 27 TShark terminal window

001	mkdir tls log	
002	cd tls_log	

Open the S_Server as previously demonstrated [Code Listing 21].

Note: Remember that we are using TPM password policy to secure the use of the private key. The password is "abc"

Code Listing 28 S_Server terminal window

001 openssl s_server -www -Verify 1 -cert ./tpm2/certs/server.crt key ./tpm2/tpm_wrapped_keys/server.key -keyform engine -engine tpm2tss -accept 127.0.0.1:8444

```
pi@raspberrypi-os:~/tpm_hardened_tls/tls_log $ openssl s_server -www -Verify 1 -cert ../tpm2/certs/server.crt -key ../tp
m2/tpm_wrapped_keys/server.key -keyform engin -engine tpm2tss -accept 127.0.0.1:8444
verify depth is 1, must return a certificate
engine "tpm2tss" set.
Enter password for user key:
Using default temp DH parameters
ACCEPT
```

Figure 29 **Opening S_Server**

On the TShark terminal window we need to start the capturing task.

Tls.pcap will be the log file created by TShark

Code Listing 29 TShark terminal window

001 tshark -s0 -w tls.pcap -i 4

pi@raspberrypi-os:~/tpm_sserver/tls_log \$ tshark -s0 -w tls.pcap -i 4 Capturing on 'Loopback'

Figure 30Start of TShark capture task

OPTIGA[™] TPM Application Note Integration of TLS Functionality for OPTIGA[™] TPM SLx 9670 TPM 2.0 Decoding SSL/TLS Traffic using TShark



In our workspace root directory (where OpenSSL conf file and self-signed certificate are located). Open the S_Client as previously demonstrated [Code Listing 22]

Code Listing 30 S_Client terminal window

```
001 openssl s_client -cert ./tpm2/certs/client.crt -key
./tpm2/tpm_wrapped_keys/client.key -keyform engine -engine tpm2tss -
connect localhost:8444
```

We will get the same interaction as in our previous section [Code Listing 22]



Figure 31 S_Client transaction

On the TShark window we will see that 13 packages were captured. Stop the TShark task as we will work with the captured file

```
pi@raspberrypi-os:~ $ tshark -s0 -w tls.pcap -i 4
Capturing on 'Loopback'
13 _
```

Figure 32 TShark captured packets

On the **tls_log** directory we will work with the captured file to get the information we want. For this we will use TShark as well.

OPTIGA[™] TPM Application Note Integration of TLS Functionality for OPTIGA[™] TPM SLx 9670 TPM 2.0 Decoding SSL/TLS Traffic using TShark



Code Listing 31 TShark terminal window

```
001 tshark -r tls.pcap -V -x -o "ssl.debug_file:ssldebug.log" -o
   "ssl.desegment_ssl_records: TRUE" -o
   "ssl.desegment_ssl_application_data: TRUE" -o
   "ssl.keys list:127.0.0.1,8444,http,server.pem" >> TLS HandShake.log
```

What the above command did is that it read the captured file with TLS handshake process and saved it to TLS_HandShake.log. The reason we do this is that the file format used by TShark is not readable by normal means.

pi@raspberrypi-os:~/tpm_hardened_tls/tls_log \$ tshark -r tls.pcap -V -x -o "ssl.debug_file:ssldebug.log" -o "ssl.desegment_ssl_records: TRUE" -o "ssl. desegment_ssl_application_data: TRUE" -o "ssl.keys_list:127.0.0.1,8444,http,server.pem" >> TLS_HandShake.log

Figure 33 Reading tls.pcap file using TShark

Now that we have the transaction between the S_Server and S_Client in a workable form we will filter the "**Secure Socket Layer**" interactions which are the ones of interest to us.

Code Listing 32 TShark terminal window

001 grep -A70 "Secure Sockets Layer" TLS_HandShake.log

These are the different steps that happened during the S_Server and S_Client TLS handshake process.



Decoding SSL/TLS Traffic using TShark

pi@raspberrypi-os:~/tpm_hardened_tls/tls_log \$ grep -A70 "Secure Sockets Layer" TLS_HandShake.log	
Secure Sockets Layer	
ILSVI Record Layer: Handshake Protocol: Client Hello	
Version: TLS 1.0 (0x0301)	
Length: 171	
Handshake Protocol: Client Hello	
length: 167	
Version: TLS 1.2 (0x0303)	
Random: fbe25a2b781d5993b5edfe79968a01f91f93093e74383b1c	
GMT Unix Time: Oct 24, 1967 11:55:39.000000000 PDT Bandem Butes: 781d5003b5edfe70068a01f01f02003e74282b1ca882bdf0	
Session ID Length: 0	
Cipher Suites Length: 56	
Cipher Suites (28 suites)	
Cipher Suite: ILS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0XC02c) Cipher Suite: ILS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0Xc030)	
Cipher Suite: TLS DHE RSA WITH AES 256 GCM SHA384 (0x009f)	
Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)	
Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)	
Cipher Suite: TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcCdd) Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xcCdd)	
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)	
Cipher Suite: TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x009e)	
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024) Cipher Suite: TLS_ECDHE_PSA_WITH_AES_256_CBC_SHA384 (0xc028)	
Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (0x006b)	
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)	
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)	
Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (0x0007) Cipher Suite: TLS ECDHE ECDSA WITH AES 256 CBC SHA (0xc00a)	
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	
Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)	
Cipher Suite: TLS_ECOHE_ECOSA_WITH_AES_128_CBC_SHA (0xc009) Cipher Suite: TLS_ECOHE RSA WITH AES_128_CBC_SHA (0xc013)	
Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)	
Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)	
Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c) Ciphen Suite: TLS_RSA_WITH_AES_266_CRC_SHA266 (0x009c)	
Cipher Suite: TLS RSA WITH AES 128 CBC SHA256 (0x005d)	
Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)	
Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)	
Compression Methods Length: 1	
Compression Methods (1 method)	
Compression Method: null (0)	
Extensions Length: 70 Extension: ec point formats (len=4)	
Type: ec_point_formats (11)	
Length: 4	
EC point formats Length: 3	
EC point format: uncompressed (0)	
EC point format: ansiX962_compressed_prime (1)	
EC point format: ansiX962_compressed_char2 (2)	
Type: supported_groups (10)	
Length: 10	
Supported Groups List Length: 8	
Supported Groups (4 groups) Supported Group: x25519 (0x001d)	
Supported Group: secp256r1 (0x0017)	
Supported Group: secp521r1 (0x0019)	
Supported Group: secp384r1 (0x0018)	
Type: SessionTicket TLS (35)	
Length: 0	
Data (0 bytes)	
Extension: encrypt_then_mac (len=0)	
Length: 0	



Decoding SSL/TLS Traffic using TShark

Secure Sockets Layer
TLSv1.2 Record Layer: Handshake Protocol: Server Hello
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 65
Handshake Protocol: Server Hello
Handshake Type: Server Hello (2)
Length: 61
Version: TLS 1.2 (0x0303)
Random: f8ed6856138131c3a65aedb96810759531ff217fac1781f5
GMT Unix Time: Mar 29, 1966 07:10:14.000000000 PST
Random Bytes: 138131C3a65aedD96810/59531t+21/tac1/815829e96bc
Session ID Length: 0
Comparts Mitte: ILS_CLUME_NSA_WIH_AES_256_GCM_SHA384 (0XC030)
Compression leader : 21
Extension: repeatiation info (len=1)
Type: renegotiation_info (65/21)
length: 1
Renegotiation Info extension
Renegotiation info extension length: 0
Extension: ec_point_formats (len=4)
Type: ec_point_formats (11)
Length: 4
EC point formats Length: 3
Elliptic curves point formats (3)
EC point format: uncompressed (0)
EC point format: ansiX962_compressed_prime (1)
EC point format: ansiX962_compressed_char2 (2)
Extension: SessionTicket TLS (len=0)
Type: SessionTicket TLS (35)
Length: 0
Data (0 bytes)
The extended master_secret (1919)
length 0
TISV1 2 Record Laver: Handshake Protocol: Certificate
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 876
Handshake Protocol: Certificate
Handshake Type: Certificate (11)
Length: 872
Certificates Length: 869
Certificates (869 bytes)
Certificate Length: 866
Certificate: 3082035e30820246020900aecba021a19d435a300d06092a (id-at-commonName=www.infineon.com,id-at-organizationalUnitName=DSS,
d-at-organizationName=Infineon,id-at-localityName=Milpitas,id-at-stateOrProvinceName=California,id-at-cou
signedCertificate
serialNumper: 12595358491/75/130
Algorithm Id: 1 2 940 113CB(1 1 11 (cha)CGUithBCAEncryntion)
Algorithm 10. 1.2.040.115345.1.1.11 (ShazSowithKSAEhCryption)
rdsequence 6 items (id-at-commonName=www infineon com id-at-organizationalUnitName=DSS id-at-organizationName=Infineon i
d-at-localitvName=Milpitas.id-at-stateOrProvinceName=California.id-at-countrvName=US)
RDNSequence item: 1 item (id-at-countrvName=US)
RelativeDistinguishedName item (id-at-countryName=US)
Id: 2.5.4.6 (id-at-countryName)
CountryName: US
RDNSequence item: 1 item (id-at-stateOrProvinceName=California)
RelativeDistinguishedName item (id-at-stateOrProvinceName=California)
Id: 2.5.4.8 (id-at-stateOrProvinceName)
DirectoryString: uTF8String (4)
uTF8String: California
RDNSequence item: 1 item (11-ar-localityName=Milpitas)
RelativeDistinguishedName item (id-at-localityName=Milpitas)
10: 2.5.4./ (10-8T-10CalltyName)
UIFECTOFYSTEINE: UIF8STEINE (4)
UIFOSTINE, "Hitplids DDNSequence item (ideat.organizationName_Infingen)
Nonsequence item: i item (iteratorganizationName=InfileOn) RelativeDistinguistendName item (iteratorganizationName=Infineon)
Id: 2.5.4.10 (id-at-organizationName)
DirectoryString: uTF8String (4)

Figure 35 TLS Server Hello

OPTIGA[™] TPM Application Note

Integration of TLS Functionality for OPTIGA[™] TPM SLx 9670 TPM 2.0

Decoding SSL/TLS Traffic using TShark

Secure Sockets Layer
TLSv1.2 Record Layer: Handshake Protocol: Certificate
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 876
Handshake Protocol: Certificate
Handshake Type: Certificate (11)
Length: 872
Certificates Length: 869
Certificates (869 bytes)
Certificate Length: 866
Certificate: 3082035530820246020900aecba021a1944359300406092a (id-at-commonName=www.infineon.com.id-at-organizationalUnitName=DSS
d-at-organizationName_Infineon_id-at-localityName_Milnitas_id-at-stateOrProvinceName_California_id-at-cou
signal artificata
carialNumber: 125053369/0177577120
set adjudinet : 12555566951/52/125
Signature (Shazowithosaenci yptun)
Argorithm 10. 1.2.040.115549.1.1.11 (Sha256withKSAEhtryption)
Issuer: runsequence (0)
ransequence: 6 items (id-at-commonwame=www.intineon.com,id-at-organizationatunitwame=uss,id-at-organizationwame=intineon,
d-at-localityName=Milpitas,id-at-stateOrProvinceName=California,id-at-countryName=US)
RDNSequence item: 1 item (id-at-countryName=US)
RelativeDistinguishedName item (id-at-countryName=US)
Id: 2.5.4.6 (id-at-countryName)
CountryName: US
RDNSequence item: 1 item (id-at-stateOrProvinceName=California)
RelativeDistinguishedName item (id-at-stateOrProvinceName=California)
Id: 2.5.4.8 (id-at-stateOrProvinceName)
DirectoryString: uTF8String (4)
uTF8String: California
RDNSequence item: 1 item (id-at-localityName=Milpitas)
RelativeDistinguishedName item (id-at-localityName=Milpitas)
Id: 2.5.4.7 (id-at-localityName)
DirectoryString: uTF8String (4)
uTF8String: Milpitas
RDNSequence item: 1 item (id-at-organizationName=Infineon)
RelativeDistinguishedName item (id-at-organizationName=Infineon)
Id: 2.5.4.10 (id-at-organizationName)
DirectoryString: uIESString (A)
Gillost Ing. InternitionalUnitName.CS)
Notsequence item. I item (id a constitutional distributions)
relativeDistinguisheuwame item (du-dc-0 gaizdatunidunitumme=DSS)
10: 2.5.4.11 (1-4-0)ganizationatonitivame)
Directorystring (4)
RUNSequence item: 1 item (id-at-commonName=WWW.infineon.com)
RelativeDistinguishedName item (id-at-commonName=www.infineon.com)
Id: 2.5.4.3 (id-at-commonName)
DirectoryString: uTF8String (4)
uTF8String: www.infineon.com
validity
notBefore: utcTime (0)
utcTime: 19-08-16 06:00:29 (UTC)
notAfter: utcTime (0)
utcTime: 20-08-15 06:00:29 (UTC)
subject: rdnSequence (0)
rdnSequence: 6 items (id-at-commonName=www.infineon.com,id-at-organizationalUnitName=DSS,id-at-organizationName=Infineon,
d-at-localityName=Milpitas,id-at-stateOrProvinceName=California,id-at-countryName=US)
RDNSequence item: 1 item (id-at-countryName=US)
RelativeDistinguishedName item (id-at-countryName=US)
Id: 2.5.4.6 (id-at-countryName)
CountryName: US
RDNSequence item: 1 item (id-at-stateOrProvinceName=California)
RelativeDistinguishedName item (id-at-stateOrProvinceName=California)
Td-2548 (id-at-stateOrProvinceName)
Directorystring (JERString (4)
ultestring (alifornia
RDNSequence item (id.at.localityName_Milnites)
NUNSequence item, i item (iterativeaticywanterniiten)
Tdt J C A Z (d at localityland)
IG: 2.5.4./ (Id-at-IOCALLYNAME)
UlrectoryString; UlrsString (4)
UI-PSSTring: Milpitas
KUNSequence item: 1 item (id-at-organizationName=Infineon)
RelativeDistinguishedName item (id-at-organizationName=Infineon)
Id: 2.5.4.10 (1d-at-organizationName)





OPTIGA[™] TPM Application Note

Integration of TLS Functionality for OPTIGA[™] TPM SLx 9670 TPM 2.0

Decoding SSL/TLS Traffic using TShark

Secure	e So	ocke	ets	Lay	/er												
TL	Sv1	1.2	Rec	oro	d La	iyer	r: ⊦	land	lsha	ake	Pro	toc	ol	: Ne	ew S	Sessio	on Ticket
	0	Cont	tent	: Ty	/pe:	На	ands	hak	(e ((22)							
	۱	/ers	sior	1: 1	ΓLS	1.2	2 (0)x03	303))							
	L	eng	gth:	10	950												
	H	land	lsha	ake	Pro	otoc	:ol:	Ne	ew S	Sess	ior	n Ti	icke	et			
			Har	ndsł	nake	е Ту	/pe:	Ne	ew S	Sess	ior	n Ti	icke	et ((4)		
			Ler	ngtł	n: 1	1046	5										
			TLS	SSE	essi	lon	Tic	ket									
				Se	essi	lon	Tic	ket	: Li	fet	ime	e Hi	int	: 72	200	secor	nds (2 hours)
				Se	essi	on	Tic	ket	: Le	engt	:h:	104	10				
				Se	essi	lon	Tic	:ket	:: f	979	9e8	d8b	08c8	3f19	9d19	afd2e	e70f880fc151d125c0cead678
TL	.Sv1	1.2	Red	oro	d La	aver	·: 0	har	ige	Cir	her	Sp	bec	Pro	otod	col: C	Change Cipher Spec
	C	Cont	tent	: T)	/pe:	Ċ	nang	e (Cipł	ner'	Spe		20)			o 1 1
	\	/ers	sior	n: 1	LS	1.2	2 (x03	303))	1						
	Length: 1																
Change Cipher Spec Message																	
TL	.Sv1	1.2	Red	ord	d La	aver	•: ⊢	land	lsha	ake	Pro	too	:01:	: Er	ncrv	/pted	Handshake Message
	C	Cont	tent	: Ty	/pe:	На	ands	hak	(e ((22)							-
	\	/ers	sior	1: 1	LS	1.2	2 (0)x03	303)								
	L	eng	gth:	40	3												
	H	land	İsha	ake	Pro	otoc	:ol:	Er	ncry	pte	ed ⊦	land	lsha	ake	Mes	ssage	
0000	00	00	00	00	00	00	00	00	00	00	00	00	08	00	45	00	E.
0010	04	86	e0	00	40	00	40	06	58	6f	7f	00	00	01	7f	00	@.@.Xo
0020	00	01	20	fc	db	0a	4f	85	61	3f	36	45	7f	b9	80	18	O.a?6E
0030	05	5d	02	7b	00	00	01	01	08	0a	b4	6c	c2	f6	b4	6c	.].{11
0040	c2	b5	16	03	03	04	1a	04	00	04	16	00	00	1c	20	04	
0050	10	f9	79	9e	8d	8b	8c	8f	19	d1	9a	fd	2e	70	f8	80	yp
0060	fc	15	1d	12	5c	0c	ea	d6	78	c4	7d	36	60	36	b8	51	\x.}6`6.Q
0070	a1	98	7f	04	са	e7	82	59	cd	33	81	4c	8c	f8	c9	4b	Y.3.LK
0080	7d	b3	fb	94	2f	cb	c7	23	ee	87	с3	cb	ab	f6	4a	c9	}/#J.
0090	c7	2c	06	23	6d	9a	ed	83	fc	f4	90	e4	ef	ba	94	14	.,.#m
00a0	0b	32	7d	40	сс	f9	4f	22	5b	ce	e5	64	25	66	35	be	.2}@O"[d%f5.
00b0	90	4e	78	4c	b7	46	88	93	fb	a9	98	36	3f	bd	89	44	.NxL.F6?D
00c0	90	d5	61	b3	f4	a8	b4	53	0a	8a	4e	da	b0	26	62	c3	aSN&b.
00d0	a7	87	53	82	ec	a4	68	e8	e6	Ød	ce	54	35	1f	d7	3d	ShT5=
00e0	fc	71	86	5f	7a	cb	11	b8	db	4d	3e	20	a4	d3	07	b3	.qzM>
00f0	73	65	8e	41	3b	7c	2a	c8	1c	e8	bd	c4	c7	3f	dd	ad	se.A; *?
0100	e5	5e	2a	3b	d8	f6	0 6	30	02	са	2e	8e	5c	03	05	e4	.^*;0\
0110	7f	13	63	6a	7d	39	88	4a	f0	94	36	fc	0d	13	8e	2f	cj}9.J6/
0120	7d	5b	47	a9	1f	68	0 8	8d	07	97	bf	33	38	f4	83	d5	}[Gh38
0130	c6	c5	46	fc	2f	5c	10	d7	bf	16	9c	56	e1	74	b3	27	F./\V.t.'
0140	2d	d5	6d	c4	9c	5c	a0	с3	16	6c	1a	fa	9b	df	eb	26	m\1&
0150	f2	ec	90	df	39	c2	55	88	94	90	3d	6a	77	b2	1f	16	9.U=jw
0160	25	9f	c4	74	82	32	09	31	4d	53	b3	37	7c	d4	29	7b	%t.2.1MS.7 .){
0170	8e	bd	b6	98	8e	56	3f	f4	06	7c	fc	e8	d5	6d	09	4f	V? m.O
0180	d1	f8	3c	79	d7	4f	80	2a	3b	70	bd	сс	32	ab	de	6f	<y.0.*;p2o< td=""></y.0.*;p2o<>
0190	1b	01	36	e7	2b	1 5	74	b2	69	26	1 9	76	a6	df	26	9d	6.+.t.1&.v&.
01a0	5d	16	5c	+1	67	6†	1 9	0b	8†	54	bd	90	a3	72	Ød	04].\.goTr
0100	63	bb	88	5e	51	40	ae	30	13	25	90	8a	60	†4	80	C5	C^QK.<.%m
0100	12	0a	ec	52	aa	a9	37	29	42	61	C1	28	D1	52	20	20	R/)BO.(.R ,
0100	58	ab	0e	65	a2	98	†3	a 6	/5	41	e9	01	80	a2	4e	CT	xeu0N.
0160	22	±3	C4	20	a4	ac	be	ee	DD	67	ca	22	95	†4	DC To	a3	,
0110	21	18	30	72 25	04	TT L	15	81	1e	60	C/	20	T/	2a	/C	60	/.=
0200	e/	99	20	21	60 47	та	27	10	a9	a2 	85	20	00	2T	29	C4	···/·· p···&·/).
0210	38	20	30	95	т/	08 -£1	er er	19	80	TT	от	00	61	12	ac	9a	8.<
0220	44	5d 66	/3	4u	0a	TI	a4	57 Fh	50	50	a9	54	04	12	cu hc	a8 1-£	
0230	1/	00	84	61	02	94	a5	50	ac Ze	T3	24	79	55	50	00	TT	a[\$yu\
0240	75	48	28	4C	00	ec	be 0e	95	70	89	44 40	80	37	aı	a0	ad	aH(Ln. /
0250	/T	TD ha	94	70	e4	T9	80		20	20	48	49	40	70	60	50	
0200	41 47	02	91	20	re £1	70	00	60 fo	ud Zd	uT c2	10	40 74	52	2a	63	59	A
0270	de	00	9/	30	TT FC	5ð	20	r d	70 f0	e2	72	/4	ري اري	17	dT	22	
0200	uC AC	20	44 28	dC E c	250	Da AA	22	00	10	9T	a 5 01	ee or	ud ef	u3 07	5e	a2 07	
0290	40 6 h	50 96	Fe	5d	ZT Ed	4u	16	Co ed	e5	24	14	62 fc	eT b0	0/	4e £1	67	L 1 2
0200	00	do	50	24	ba	CT 20	10	eu	ED	24	14 0	06	00	ou e0	11	20	- 043 *
0200	12	he	20	∠u fo	62	64	52	0T	57 -£1	Oh	es EE	20	61	69	ar	2a 10	
0200	12	96	20	20	12	44	55 1-F	24	52	75	22	24	00	57 6f	10	+9	9B 7U - OTW
0200	10	55	06	56	42	22	25	co	90	66	7f	5u	77	30	49	67	
02E0	16	62	74	62	50	20	44	22	62	h/	44	90	47	h/	e/	h7	
5210	10	-22	<i>,</i> u	ca	Ju	20	uu	ud	00	0+	+	26	/	0+	C+	07	

Figure 37

TLS Creation of Session Ticket : All communication are encrypted at this point





TShark is a useful tool that enables engineers and hackers to monitor transactions in a network. The important fact of using OPTIGA[™] SLx 9670 TPM 2.0 as hardening element of a TLS session is that the private key used to enable the session is guarded by OPTIGA[™] SLx 9670 TPM 2.0. The use of the Private Key is also protected by a TPM 2.0 password policy. More complex policies can be created around TPM 2.0 objects. These are out of the scope of this application note.

As we have mentioned the TLS layer sits on top of the Transport Layer and provides the means of encrypting the communication channel between two entities. Additionally, TLS is a point to point transaction and not end to end. In other words, the TLS channel is established between two parties.

A higher level of security can be achieved by implementing Token Binding which is an extension to TLS. Token Binding Protocol is established by the user-agent generating a Private Key-Public Key pair (managed by TPM) per target server, and proving possession of the private key on every TLS connection to the target server.



4 References

- [1] http://www.infineon.com/tpm
- [2] <u>https://github.com/tpm2-software</u>
- [3] https://www.openssl.org/
- [4] <u>https://trustedcomputinggroup.org/work-groups/trusted-platform-module/</u>
- [5] https://trustedcomputinggroup.org/work-groups/software-stack/
- [6] https://trustedcomputinggroup.org/wp-content/uploads/TSS_ESAPI_Version-0.9_Revision-04_reviewEND030918.pdf
- [7] https://www.raspberrypi.org/downloads/raspbian/
- [8] <u>https://www.wireshark.org/</u>
- [9] <u>https://www.rfc-editor.org/rfc/rfc8472.txt</u>
- [10] https://www.globalsign.com/en/



Revision history

Document version	Date of release	Description of changes
1.0	27.04.2020	Initial version

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2020-04-27

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2020 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document? Email:

dsscustomerservice@infineon.com

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of noninfringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application. For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.