

Computer and Network Security Project

Ufuk Çağatay DOĞAN
Şevki Armağan OĞUZ
Seyit Bahadır ÇETİN
Berk ÖNDER

Table of Contents

1. Part of The Project: Creating a free certificate and Thunderbird certificate loading	<hr/> 2
ÇAĞATAY DOĞAN	<hr/> 2
BERK ÖNDER	<hr/> 5
BAHADIR ÇETİN	<hr/> 8
ARMAĞAN OĞUZ	<hr/> 11
2. Part of The Project: Sending a mail by forwarding each other PEM certificates	<hr/> 14
Çağatay DOĞAN Screenshots:	<hr/> 14
Berk ÖNDER Screenshots:	<hr/> 16
Armağan OĞUZ Screenshots:	<hr/> 18
3. Part of The Project: Configuring Apache Webserver	<hr/> 19
Çağatay DOĞAN Screenshots:	<hr/> 19
Bahadır ÇETİN Screenshots:	<hr/> 22
Berk ÖNDER Screenshots:	<hr/> 24
Armağan OĞUZ Screenshots:	<hr/> 26
4. Part of The Project: Encrypting an Image and Modifying AES.java	<hr/> 29
Encrypting an Image	<hr/> 29
Çağatay Screenshots:	<hr/> 29
Berk ÖNDER Screenshots:	<hr/> 31
Bahadir Çetin Screenshots:	<hr/> 33
Armağan OĞUZ Screenshots:	<hr/> 35
Modifying AES.java	<hr/> 37
Çağatay DOĞAN Screenshots:	<hr/> 37
Bahadır ÇETİN Screenshots:	<hr/> 40
Berk ÖNDER Screenshots:	<hr/> 43
Armağan OĞUZ Screenshots:	<hr/> 47
5. Part of The Project: Secure Code Standards	<hr/> 50
References:	<hr/> 51
SVG Images:	<hr/> 51

1. Part of The Project: Creating a free certificate and Thunderbird certificate loading

ÇAĞATAY DOĞAN

Name Surname: Ufuk Çağatay Doğan

Encryption Algorithm: RSA

Hash Algorithm: SHA-256 with RSA Encryption

Key Length: 2048

Validity:

Not After 12/15/2021, 11:22:38 PM (GMT+03:00)

Not After 12/15/2021, 11:22:38 PM (GMT+03:00)

Serial Number: 15:03:85:35:36:05:87:B8:E5:78:38:A3:AC:08:BE:4F

Issuer Name:

Country: IT

State/Province: Bergamo

Locality: Ponte San Pietro

Organization: Actalis S.p.A.

Common Name: Actalis Client Authentication CA G3

Public Key:

Algorithm RSA

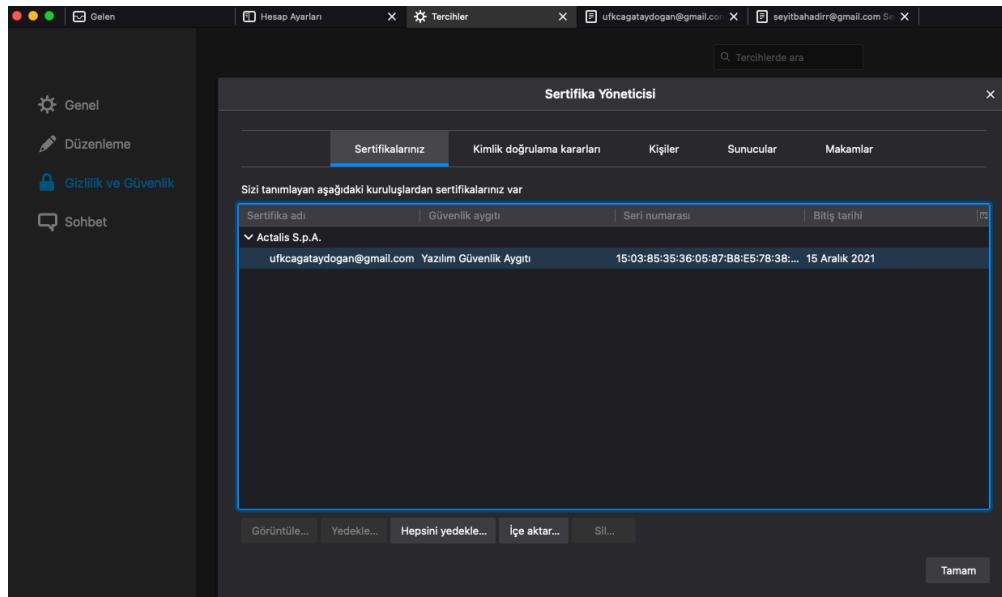
Key Size 2048

Exponent 65537

Modulus

92:3B:80:3B:28:1F:88:D2:3A:07:78:52:4D:69:94:10:EB:FE:08:58:55:C2:F2:FA:31:56:6F:6D
:6C:6C:26:B8:81:0C:87:4F:F7:73:AF:41:48:D5:45:68:BC:19:94:3C:F9:36:9C:BF:CA:89:87:
3B:BC:B8:3C:EB:78:A2:60:CA:17:10:42:30:FD:6E:6F:B9:07:EC:A8:76:A0:4A:24:9F:DF:1
2:D6:F7:54:5B:B1:22:51:7B:DB:4C:0C:D1:C4:09:36:04:28:96:11:68:A1:C5:B5:F8:51:70:42:
76:31:4A:E5:E4:84:6A:F8:69:DB:AE:A3:5E:26:37:73:08:30:DE:B1:AA:7A:C1:7F:65:97:23:
37:99:3F:79:2D:F5:E9:91:E8:08:BD:D3:27:B3:5D:A5:95:3D:F2:BB:E0:11:DD:D1:F8:C8:A
6:E5:AA:B1:B0:BC:88:7B:7D:B0:C5:F8:FE:54:4C:7C:1D:81:5A:B7:5E:08:92:D9:B8:F9:31:
4D:9D:04:C4:86:EC:26:FF:3B:05:21:19:56:2F:C6:3E:38:9F:70:B6:F5:DD:0E:40:93:66:C0:D
A:8C:84:7F:F2:85:E6:C0:40:DE:9A:3F:C0:34:06:CB:28:0C:2B:0B:58:AD:4D:7F:E8:B2:BD:
D2:24:F2:67:57:9D:E0:A2:F6:02:B8:A8:7D

ÇAĞATAY Screenshots:



Sertifika Yöneticisi

Sertifikalarınız

Sizi tanımlayan aşağıdaki kuruluşlardan sertifikalarınız var

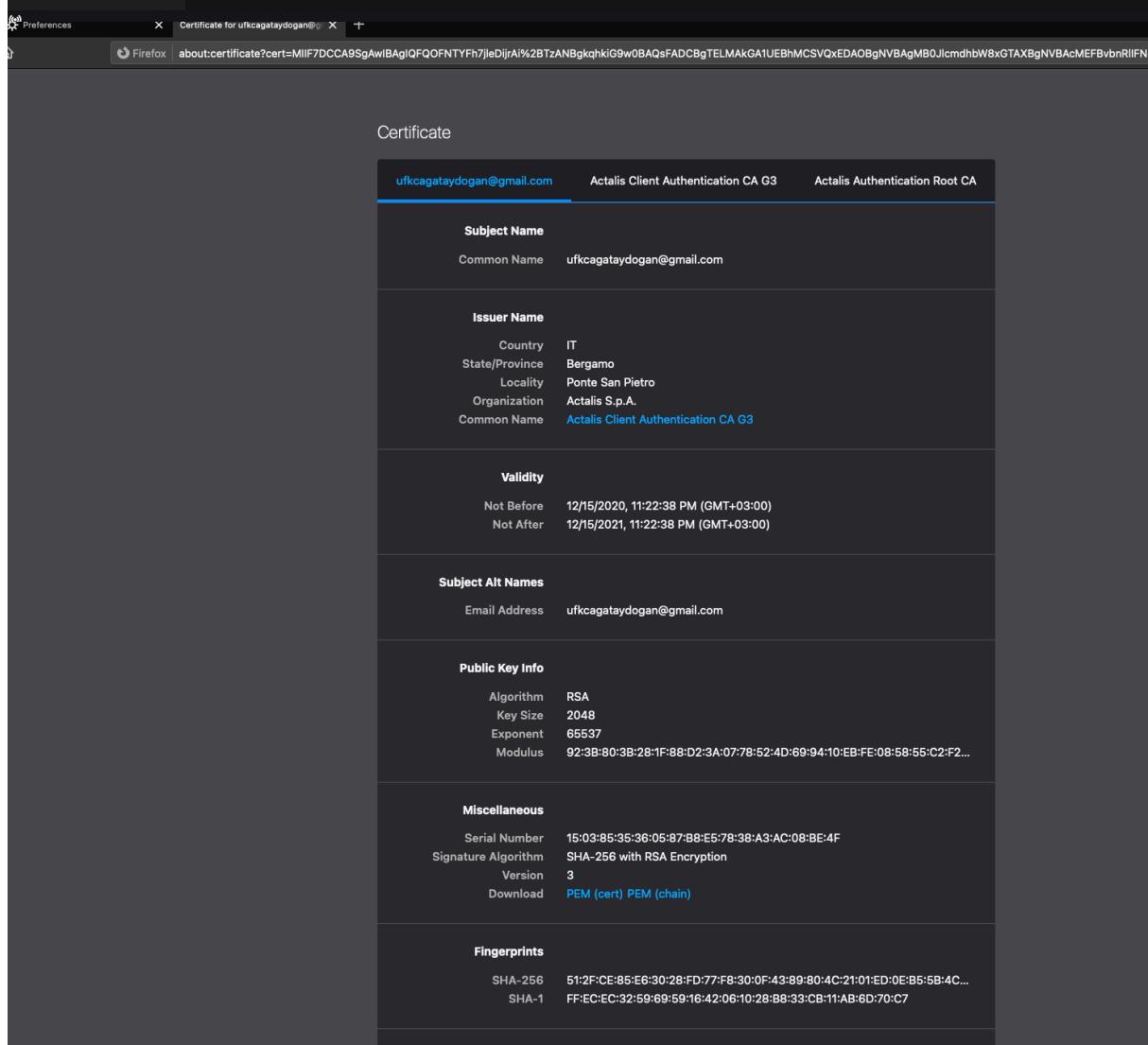
Sertifika adı: ufkcagataydoran@gmail.com | Güvenlik aygıtı: Yazılım Güvenlik Aygıtı | Seri numarası: 15:03:85:35:36:05:87:B8:E5:78:38... | Bitiş tarihi: 15 Aralık 2021

Görüntüle... Yedekle... Hepşini yedekle... içe aktar... Sil... Tamam

Actalis S.p.A.

ufkcagataydoran@gmail.com Yazılım Güvenlik Aygıtı 15:03:85:35:36:05:87:B8:E5:78:38... 15 Aralık 2021

Eklentiler ve temalar



Certificate

uftkcagataydoran@gmail.com Actalis Client Authentication CA G3 Actalis Authentication Root CA

Subject Name

Common Name: uftkcagataydoran@gmail.com

Issuer Name

Country: IT
State/Province: Bergamo
Locality: Ponte San Pietro
Organization: Actalis S.p.A.
Common Name: Actalis Client Authentication CA G3

Validity

Not Before: 12/15/2020, 11:22:38 PM (GMT+03:00)
Not After: 12/15/2021, 11:22:38 PM (GMT+03:00)

Subject Alt Names

Email Address: uftkcagataydoran@gmail.com

Public Key Info

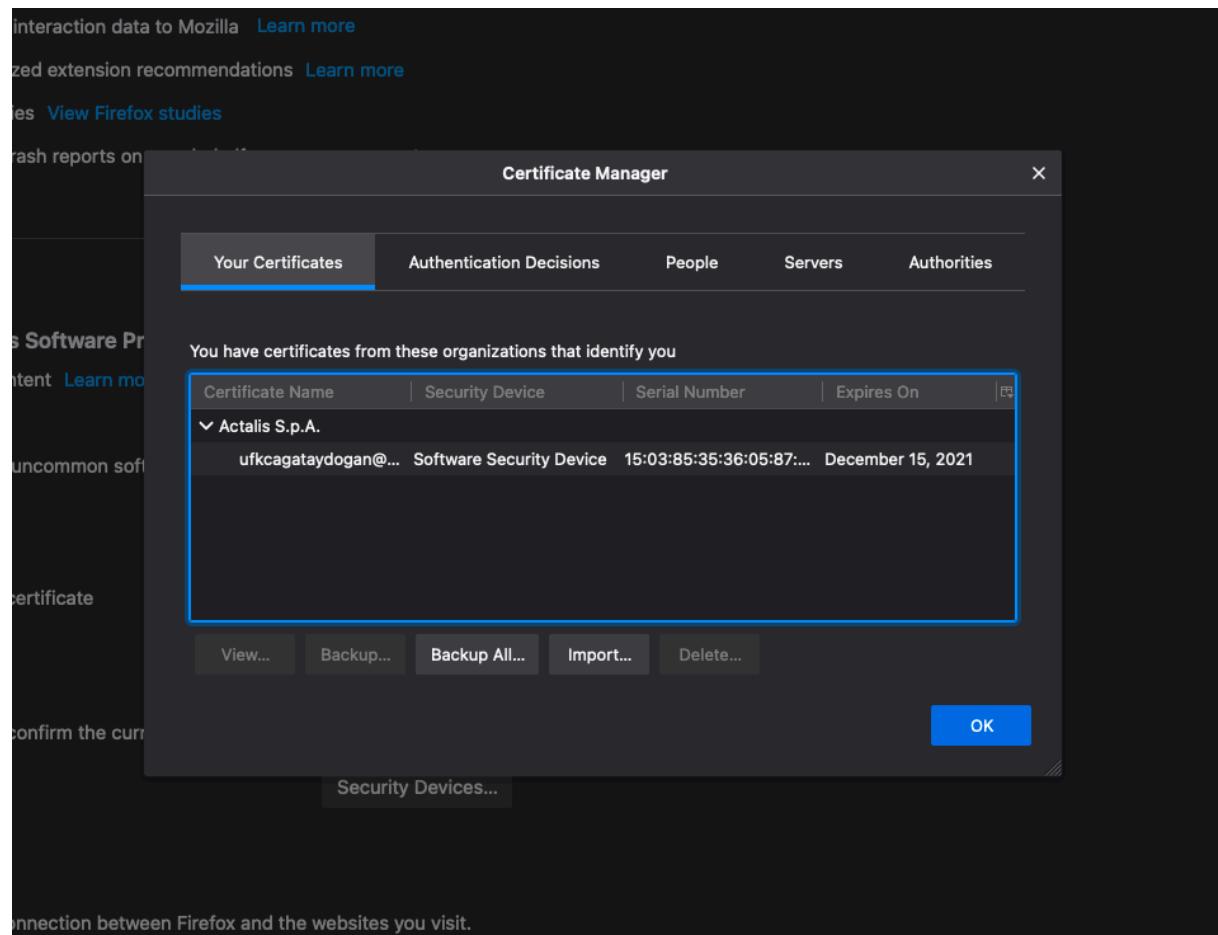
Algorithm: RSA
Key Size: 2048
Exponent: 65537
Modulus: 92:3B:80:3B:28:1F:88:D2:3A:07:78:52:4D:69:94:10:EB:FE:08:58:55:C2:F2...

Miscellaneous

Serial Number: 15:03:85:35:36:05:87:B8:E5:78:38:A3:AC:08:BE:4F
Signature Algorithm: SHA-256 with RSA Encryption
Version: 3
Download: PEM (cert) PEM (chain)

Fingerprints

SHA-256: 51:2F:CE:85:E6:30:28:FD:77:F8:30:0F:43:89:80:4C:21:01:ED:0E:B5:5B:4C...
SHA-1: FF:EC:32:59:69:59:16:42:06:10:28:B8:33:CB:11:AB:6D:70:C7



BERK ÖNDER

Name Surname: Berk Önder

Encryption Algorithm: RSA

Hash Algorithm: SHA-256 with RSA Encryption

Key Length: 2048

Validity:

Not Before 12/15/2020, 11:21:08 PM (GMT+03:00)

Not After 12/15/2021, 11:21:08 PM (GMT+03:00)

Serial Number: 12:0C:2F:FF:5B:E1:6B:CA:A3:80:6C:46:4D:81:0B:43

Issuer Name:

Country: IT

State/Province: Bergamo

Locality: Ponte San Pietro

Organization: Actalis S.p.A.

Common Name: Actalis Client Authentication CA G3

Public Key:

Algorithm RSA

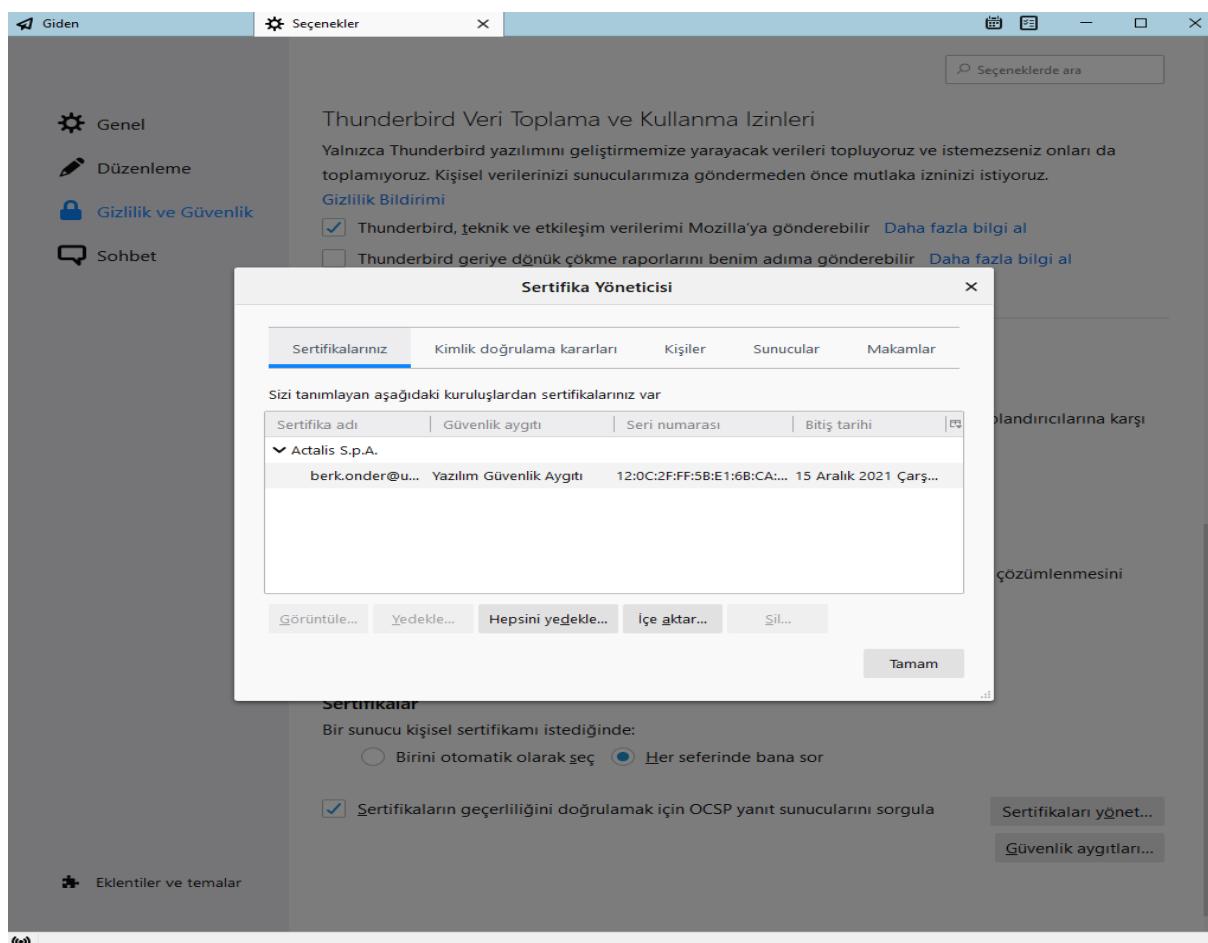
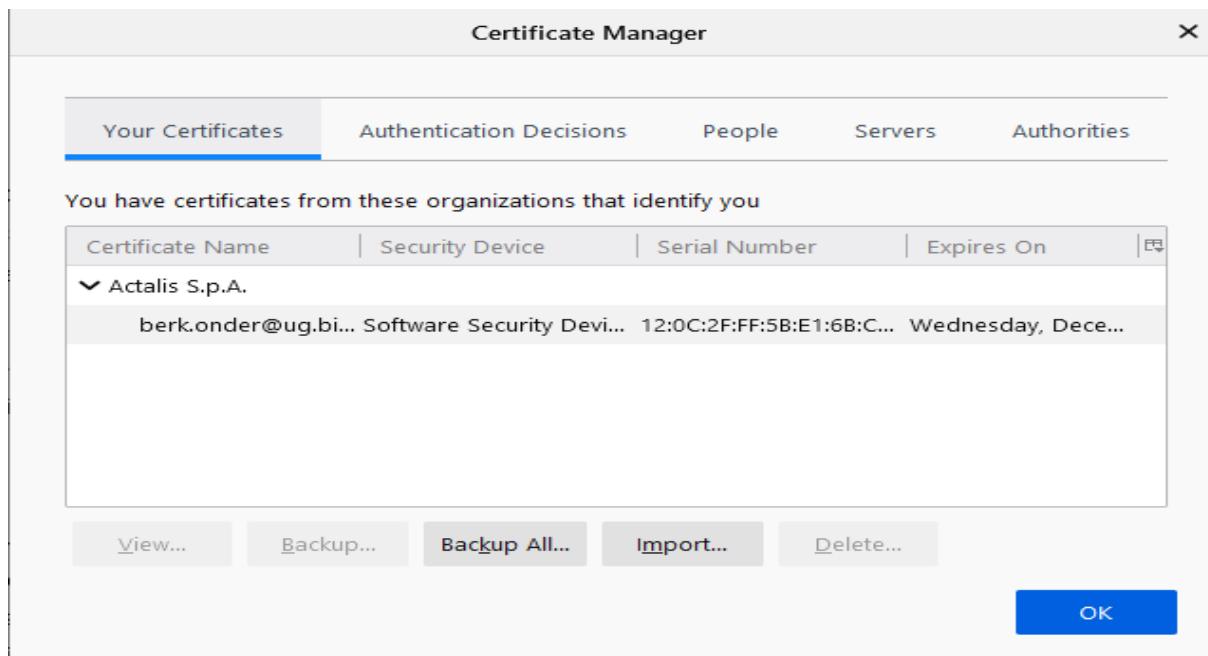
Key Size 2048

Exponent 65537

Modulus

B8:6B:A8:D4:69:5B:53:06:63:4D:72:37:0D:10:75:02:1D:A2:6D:30:5B:B3:55:FA:76:FE:06:
11:40:A7:4C:AA:A7:3F:F6:0F:F0:8A:B1:83:A1:24:13:9D:ED:2F:78:4A:FE:1A:E9:BB:F5:C
1:19:04:FD:DF:02:00:0E:EA:69:51:A4:1A:0A:03:44:D9:0A:15:CA:98:1A:BA:58:86:67:7D:
DE:87:70:BC:A7:55:16:BD:6B:B9:AA:80:1D:BD:B6:A7:F1:01:17:18:0C:81:5F:C3:F7:BC:2
4:11:77:2C:D5:37:26:47:7C:4A:60:A2:A3:DA:F5:79:AF:1F:47:42:F5:17:96:F3:76:E5:11:81:
C0:6F:C7:BC:B0:34:2D:86:D0:06:1A:89:94:9A:B4:08:EA:A1:B9:8A:6C:72:F0:52:99:66:B0:
D6:74:9E:06:B9:F3:02:35:47:D2:56:86:63:2C:74:93:17:C4:9E:28:EB:49:A3:F3:6F:BC:30:A
9:80:DC:7A:29:50:2A:E3:26:B0:FA:34:4E:3E:0B:25:60:FF:3A:6F:33:80:8D:81:9F:48:31:D0
:26:DC:4D:15:F2:AD:8A:CD:0B:87:3E:53:AA:5A:70:F1:E8:81:0C:8A:61:92:31:B4:DB:12:5
9:3B:E7:F2:2E:B2:95:CE:59:EE:3D:74:B2:A9

BERK ÖNDER Screenshots:



Certificate

berk.onder@ug.bilkent.edu.tr

Actalis Client Authentication CA G3

Actalis Authentication Root CA

Subject Name

Common Name berk.onder@ug.bilkent.edu.tr

Issuer Name

Country IT
State/Province Bergamo
Locality Ponte San Pietro
Organization Actalis S.p.A.
Common Name [Actalis Client Authentication CA G3](#)

Validity

Not Before 12/15/2020, 11:21:08 PM (GMT+03:00)
Not After 12/15/2021, 11:21:08 PM (GMT+03:00)

Subject Alt Names

Email Address berk.onder@ug.bilkent.edu.tr

Public Key Info

Algorithm RSA
Key Size 2048
Exponent 65537
Modulus B8:6B:A8:D4:69:5B:53:06:63:4D:72:37:0D:10:75:02:1D:A2:6D:30:5B:B3:55:FA:76:FE:06:11:4...

Miscellaneous

Serial Number 12:0C:2F:FF:5B:E1:6B:CA:A3:80:6C:46:4D:81:0B:43
Signature Algorithm SHA-256 with RSA Encryption
Version 3
Download [PEM \(cert\)](#) [PEM \(chain\)](#)

Fingerprints

SHA-256 2C:7B:AA:DC:A5:28:6E:8A:A8:D2:F6:3B:C2:AC:98:DB:FB:5F:DF:04:C0:70:93:93:5D:11:CF:4...
SHA-1 1F:09:6F:26:B7:9B:C3:BF:1B:BA:05:B6:4A:92:6F:7E:D7:4C:59:27

BAHADIR ÇETİN

Name Surname: Seyit Bahadir Çetin

Encryption Algorithm: RSA

Hash Algorithm: SHA-256 with RSA Encryption

Key Length: 2048

Validity:

Not Before 12/15/2020, 10:59:12 PM (GMT+03:00)

Not After 12/15/2021, 10:59:12 PM (GMT+03:00)

Serial Number: 54:0A:B6:49:E0:0F:43:C5:F3:39:37:99:0F:1E:93:0D

Issuer Name:

Country: IT

State/Province: Bergamo

Locality: Ponte San Pietro

Organization: Actalis S.p.A.

Common Name: Actalis Client Authentication CA G3

Public Key:

Algorithm RSA

Key Size 2048

Exponent 65537

Modulus

F2:60:A9:F8:32:18:80:36:18:8E:33:97:E1:E3:44:43:8A:09:9D:BB:0A:FE:49:E7:75:CD:14:8
0:6E:4B:85:5D:05:3C:EA:79:C7:DD:02:8D:14:BC:23:CE:42:44:6F:72:10:19:65:9A:9F:8B:7
A:50:D9:49:95:55:39:68:12:F7:B0:FC:96:3F:DA:60:9E:54:88:A5:99:7C:08:E8:86:00:C8:52:
6B:45:10:A1:5A:F0:3F:7E:B6:3F:D9:B5:C0:76:0D:DB:B9:4E:9C:D8:04:BE:DF:D4:7B:AF:
3B:B9:F3:6D:44:7E:26:EB:B7:D4:AA:13:41:93:DB:DE:44:67:7E:A7:8E:7D:01:42:9D:B7:7
A:69:4A:35:AF:BA:18:1D:E0:83:8E:5F:B1:44:7F:49:E4:05:1B:DC:5D:D6:1A:61:07:2E:3B:
54:E5:BA:B1:35:12:2E:4E:99:CD:FE:29:34:8A:D2:9D:06:F6:E6:97:FD:D5:63:04:FE:E0:5D:
15:2E:DB:59:7A:03:B7:8E:28:1B:98:C8:67:3D:35:07:5D:48:C3:2E:6D:B7:2F:5E:3E:A3:87:
86:EE:CF:4C:AD:6C:C9:FC:2C:57:A3:FB:EC:39:A4:8D:DE:A1:FD:79:F4:95:A2:C4:5C:B0:
2D:CF:E3:AD:41:85:5B:4C:70:40:0D:E8:99:93:47

Bahadir ÇETİN Screenshots:

The screenshot shows the "Sertifika Yöneticisi" (Certificate Manager) window in Thunderbird. The "Sertifikalarınız" tab is selected. It displays a certificate issued by "Actalis S.p.A." to "seyitbahadirr@g...". The certificate details are: "Güvenlik aygıtı" (Security Device) - "seyitbahadirr@g...", "Seri numarası" (Serial Number) - "54:0A:B6:49:E0:0F:43:C5...", and "Bitiş tarihi" (Expiration Date) - "15 Aralık 2021 Çarş...". Below the table are buttons: "Görüntüle..." (View), "Yedekle..." (Backup), "Hepsini yedekle..." (Backup All...), "İçe aktar..." (Import), and "Sil..." (Delete). A "Tamam" (OK) button is at the bottom right.

The screenshot shows the "Certificate Manager" window. The "Your Certificates" tab is selected. It displays a certificate issued by "Actalis S.p.A." to "seyitbahadirr@g...". The certificate details are: "Security Device" - "Software Security Devi...", "Serial Number" - "54:0A:B6:49:E0:0F:43:C5...", and "Expires On" - "Wednesday, Decem...". Below the table are buttons: "View...", "Backup...", "Backup All...", "Import...", and "Delete...". At the bottom right is an "OK" button. A note at the bottom left says "Query OCSP responder servers to confirm the current validity of certificates" and links to "View Certificates..." and "Security Devices...".

Certificate

seyitbahadir@gmail.com	Actalis Client Authentication CA G3	Actalis Authentication Root CA
Subject Name		
Common Name	seyitbahadir@gmail.com	
Issuer Name		
Country	IT	
State/Province	Bergamo	
Locality	Ponte San Pietro	
Organization	Actalis S.p.A.	
Common Name	Actalis Client Authentication CA G3	
Validity		
Not Before	12/15/2020, 10:59:12 PM (GMT+03:00)	
Not After	12/15/2021, 10:59:12 PM (GMT+03:00)	
Subject Alt Names		
Email Address	seyitbahadir@gmail.com	
Public Key Info		
Algorithm	RSA	
Key Size	2048	
Exponent	65537	
Modulus	F2:60:A9:F8:32:18:80:36:18:8E:33:97:E1:E3:44:43:8A:09:9D:B8:0A:FE:49:E7:75:CD:1...	
Miscellaneous		
Serial Number	54:0A:B6:49:E0:0F43:C5:F3:39:37:99:0F:1E:93:0D	
Signature Algorithm	SHA-256 with RSA Encryption	
Version	3	
Download	PEM (cert) PEM (chain)	
Fingerprints		
SHA-256	DA:8C:58:D0:B0:C2:80:5A:FA:60:B7:32:0E:84:5A:2C:43:32:CB:7E:82:CF:BE:53:55:A2...	
SHA-1	83:1A:99:80:CE:FF:FA:D3:6A:EF:1E:28:5B:E4:F3:18:C5:B0:DD:DA	
Basic Constraints		

ARMAĞAN OĞUZ

Name Surname: Armağan Oğuz

Encryption Algorithm: RSA

Hash Algorithm: SHA-256 with RSA Encryption

Key Length: 2048

Validity:

Not Before 17.12.2020 15:24:25 (GMT+03:00)

Not After 17.12.2021 15:24:25 (GMT+03:00)

Serial Number: 1E:C6:DF:48:0E:D6:AA:10:F4:6A:17:BA:E3:36:45:D6

Issuer Name:

Country: IT

State/Province: Bergamo

Locality: Ponte San Pietro

Organization: Actalis S.p.A.

Common Name: Actalis Client Authentication CA G3

Public Key:

Algorithm RSA

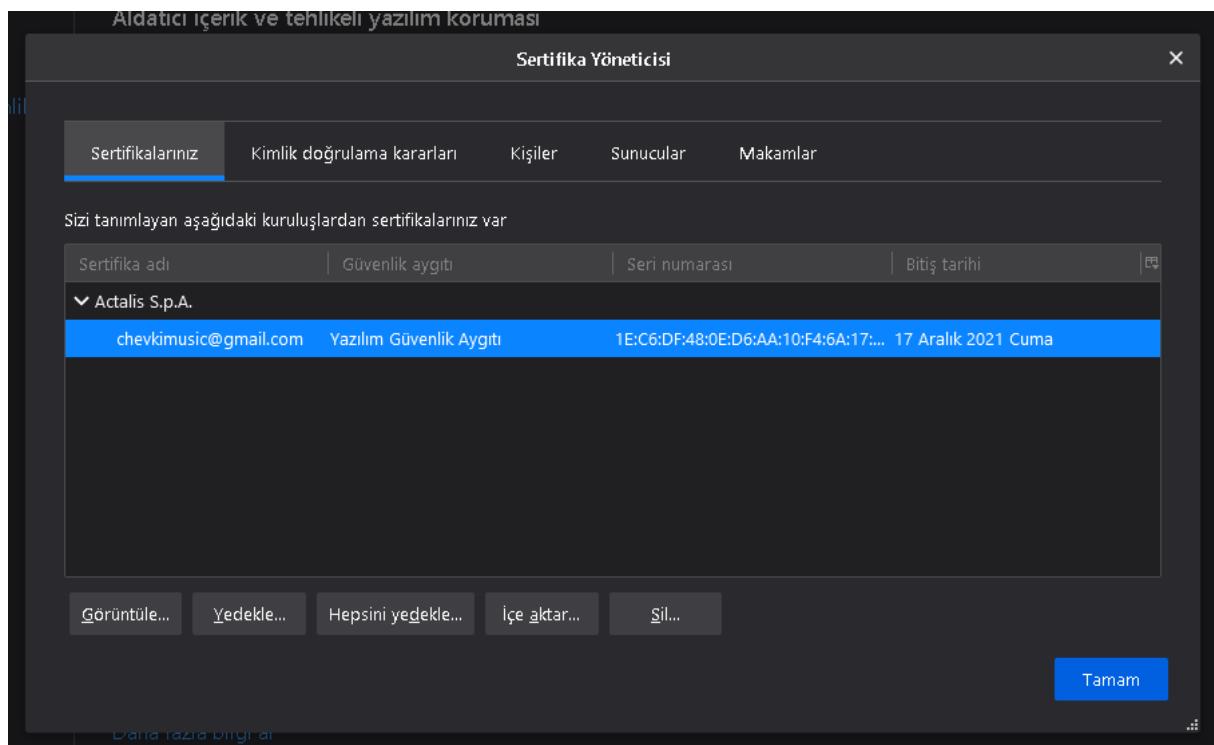
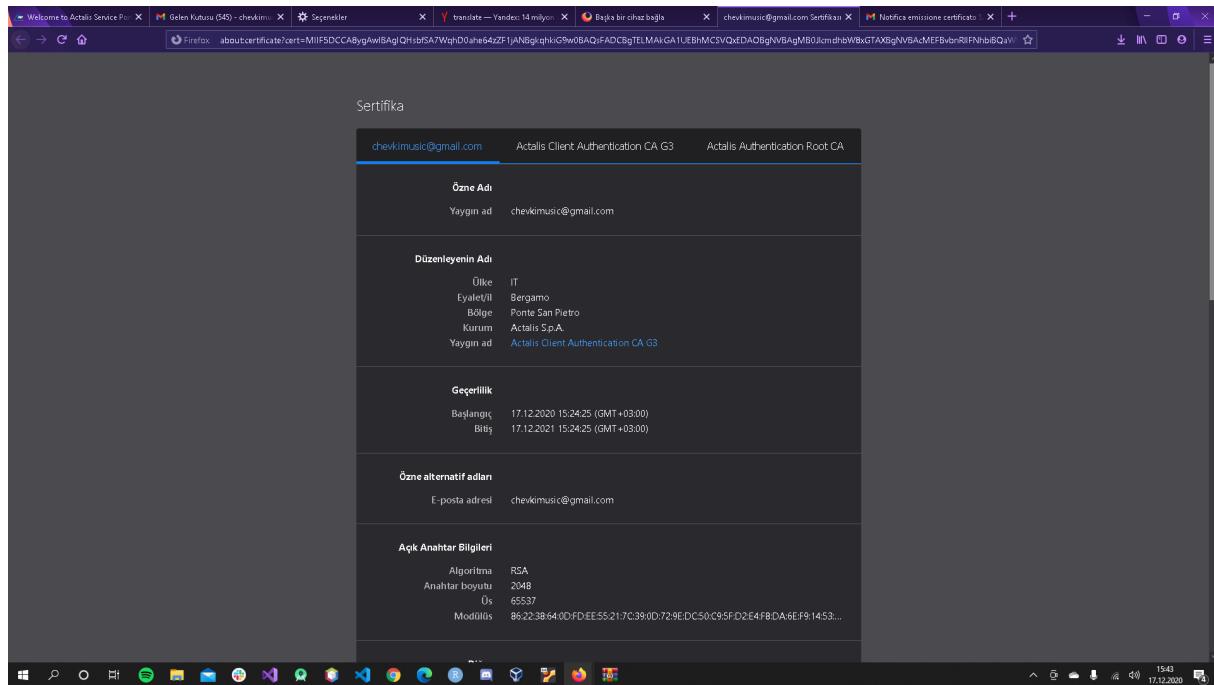
Key Size 2048

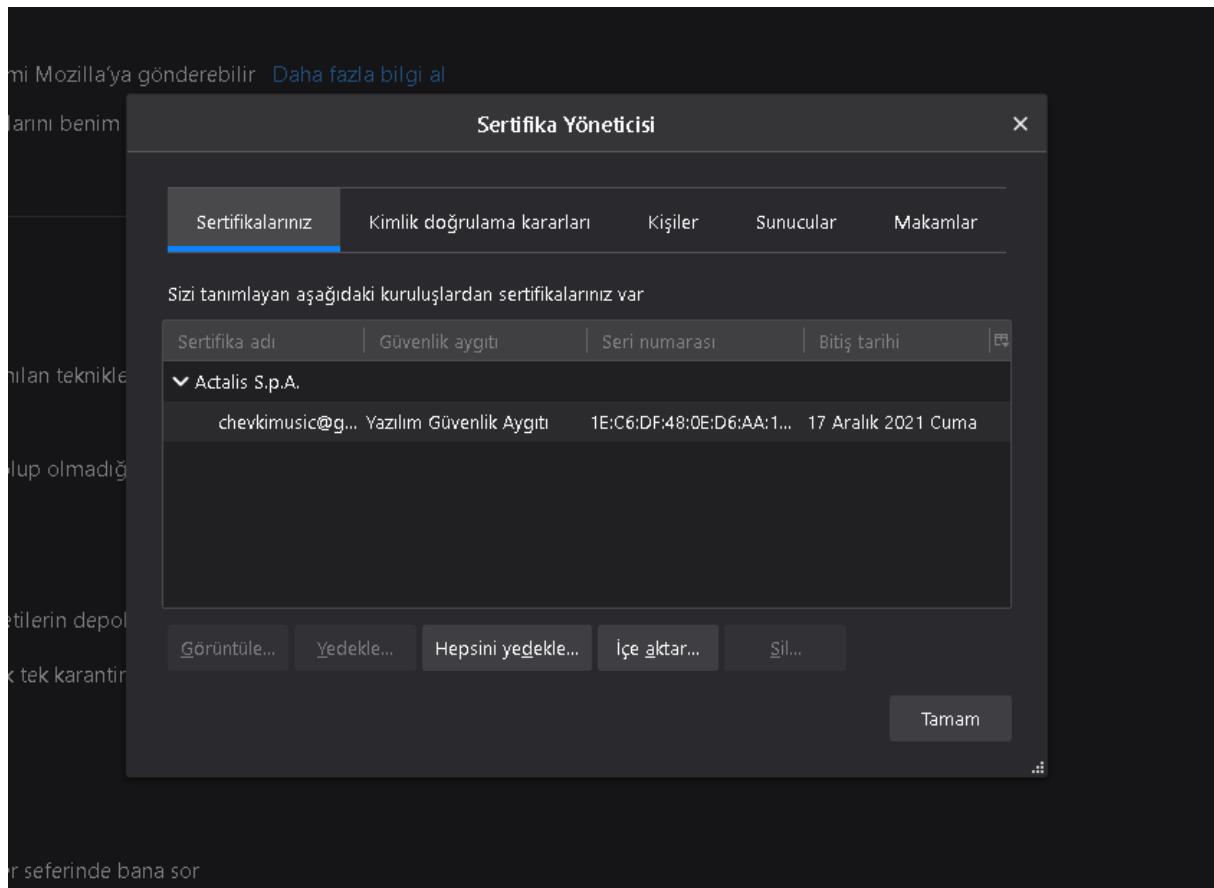
Exponent 65537

Modulus

86:22:38:64:0D:FD:EE:55:21:7C:39:0D:72:9E:DC:50:C9:5F:D2:E4:F8:DA:6E:F9:14:53:B2:
F8:A6:DF:B3:10:72:C7:BA:AD:DE:D9:63:8D:44:96:4E:37:13:38:22:C3:E8:25:E4:E0:15:38:
77:65:B1:B8:B0:20:30:BA:F4:CA:D3:10:B5:89:B7:62:4E:2B:C7:66:6C:10:8E:BE:65:6F:D2:
B9:34:58:4F:CD:3F:7B:67:FF:F0:C5:98:03:88:BC:7D:B8:EE:98:5E:C8:34:F3:3B:63:D0:2D:
8D:D4:AE:5F:2E:62:61:0B:20:8D:BB:0F:69:D2:8C:96:B4:A6:5B:E2:BF:6E:1D:33:30:7B:61:
:38:36:F9:95:D0:C1:C9:29:FF:FE:E8:AC:8A:CA:9B:C3:20:C1:72:5A:3A:FE:13:76:00:47:4F:
:68:56:BE:DF:7B:92:06:29:09:8D:C0:16:AF:6E:90:22:F7:63:B8:AD:31:8D:EF:00:93:35:C7:
00:6B:52:D1:A6:1A:C7:C4:6F:BF:F7:F5:80:36:7A:4F:B8:97:76:C6:47:F2:93:B8:94:0F:B9:8
4:E7:F1:69:F5:38:D1:D1:D2:C0:15:29:08:66:5E:C4:9C:64:2F:28:D6:69:44:C5:83:70:EE:39:
89:C2:BC:8B:F8:85:8C:42:5E:65:A2:75

Armağan OĞUZ Screenshots:





As we created our S/MIME certificates which are free from Actalis, then we can use them while we are sending encrypted e-mails via Mozilla Thunderbird. By doing this, our e-mails are secure, signed, and encrypted now. After we send e-mails to each other, we observed these e-mails cannot be seen on other platforms such as gmail.com. They can be only seen on Thunderbird.

The details of the certificate issuer can be seen on Mozilla Thunderbird by doing following steps: Options-->Privacy&Security-->Certificates-->Manage Certificates-->Your Certificates-->Double Click on the Certificate-->Go to Issuer Name Part (Click on the Common Name). Now you can check the certificate authority details.

2. Part of The Project: Sending a mail by forwarding each other PEM certificates

Çağatay DOĞAN Screenshots:

The screenshot shows the 'Sertifika Yöneticisi' (Certificate Manager) window in Thunderbird. The 'Kişiler' (People) tab is selected. A table lists certificates for 'Actalis S.p.A.' with columns for 'Sertifika adı' (Certificate name), 'E-posta adresi' (Email address), and 'Bitiş tarihi' (Expiration date). Three entries are shown:

Sertifika adı	E-posta adresi	Bitiş tarihi
berk.onder@ug.bilkent.edu.tr	berk.onder@ug.bilkent.edu.tr	15 Aralık 2021
chevkimusic@gmail.com	chevkimusic@gmail.com	17 Aralık 2021
seyitbahadir@gmail.com	seyitbahadir@gmail.com	15 Aralık 2021

At the bottom left, there is a checked checkbox for 'Sertifikaların geçerliliğini doğrulamak için OCSP yanıt sunucularını sorgula' (Check for OCSP responder servers to verify certificate validity). At the bottom right, there are buttons for 'Sertifikaları yönet...' (Manage certificates...) and 'Güvenlik aygıtları...' (Security devices...).

The screenshot shows an email message in Thunderbird. The message is from 'Çağatay Doğan' to '496 Project'. The subject is '496 Project'. The message body says 'Merhaba ben Çağatay Doğan.' A detailed security header is visible at the top of the message area, showing:

İleti güvenliği - S/MIME
İleti imzalanmış
Bu iletide geçerli bir sayısal imza var. İleti gönderildikten sonra değiştirilmemiş.
İzleyen: ufkagataydogan@gmail.com
E-posta adresi: ufkagataydogan@gmail.com
Sertifikayı sağlayan: Actalis Client Authentication CA G3
İmza sertifikasını göster

İleti şifrelenmiş
Bu ileti size göndermeden önce şifrelenmiş. Şifreleme, bu iletiyi açmadan önce gerekli olan şifrelerle okunmasını sağlar.

Bahadir ÇETİN Screenshots:

Sertifika Yöneticisi

Sertifikalarınız Kimlik doğrulama kararları **Kişiler** Sunucular Makamlar

Aşağıdaki kişileri tanımlayan kayıtlı sertifikalarınız var

Sertifika adı	E-posta adresi	Bitiş tarihi
Actalis S.p.A.	berk.onder@ug.bilkent.... berk.onder@ug.bilkent.edu.tr	15 Aralık 2021 Çarşamba
	chevkimusic@gmail.com chevkimusic@gmail.com	17 Aralık 2021 Cuma
	ufkcagataydogan@gma... ufcagataydogan@gmail.com	15 Aralık 2021 Çarşamba

Görüntüle... İçe aktar... Dışa aktar... Sil... Tamam

Gonderen **Benden** Yanıtla Tümüne yanıtla İlet Arşivle Gereksiz Sil Daha fazla 17.12.2020 16:11 S/MIME

Hello it's an encrypted mail by using S/MIME.
Seyit Bahadir Cetin

İleti güvenliği - S/MIME

İleti imzalanmış
Bu iletide geçerli bir sayısal imza var. İleti gönderildikten sonra değiştirilmemiş.

İmzlayan: seyitbahadir@gmail.com
E-posta adresi: seyitbahadir@gmail.com
Sertifikayı sağlayan: Actalis Client Authentication CA G3

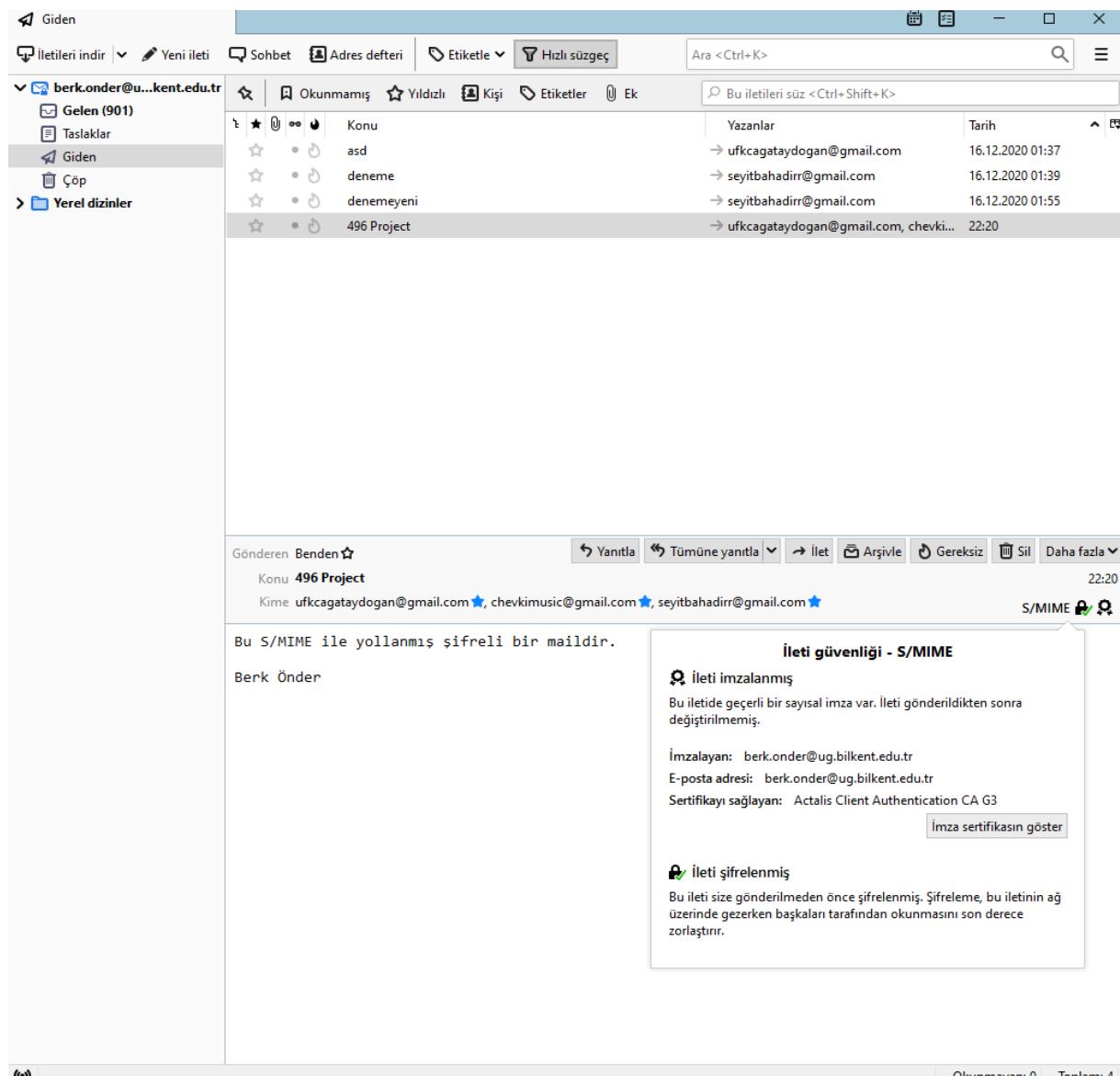
İmza sertifikasını göster

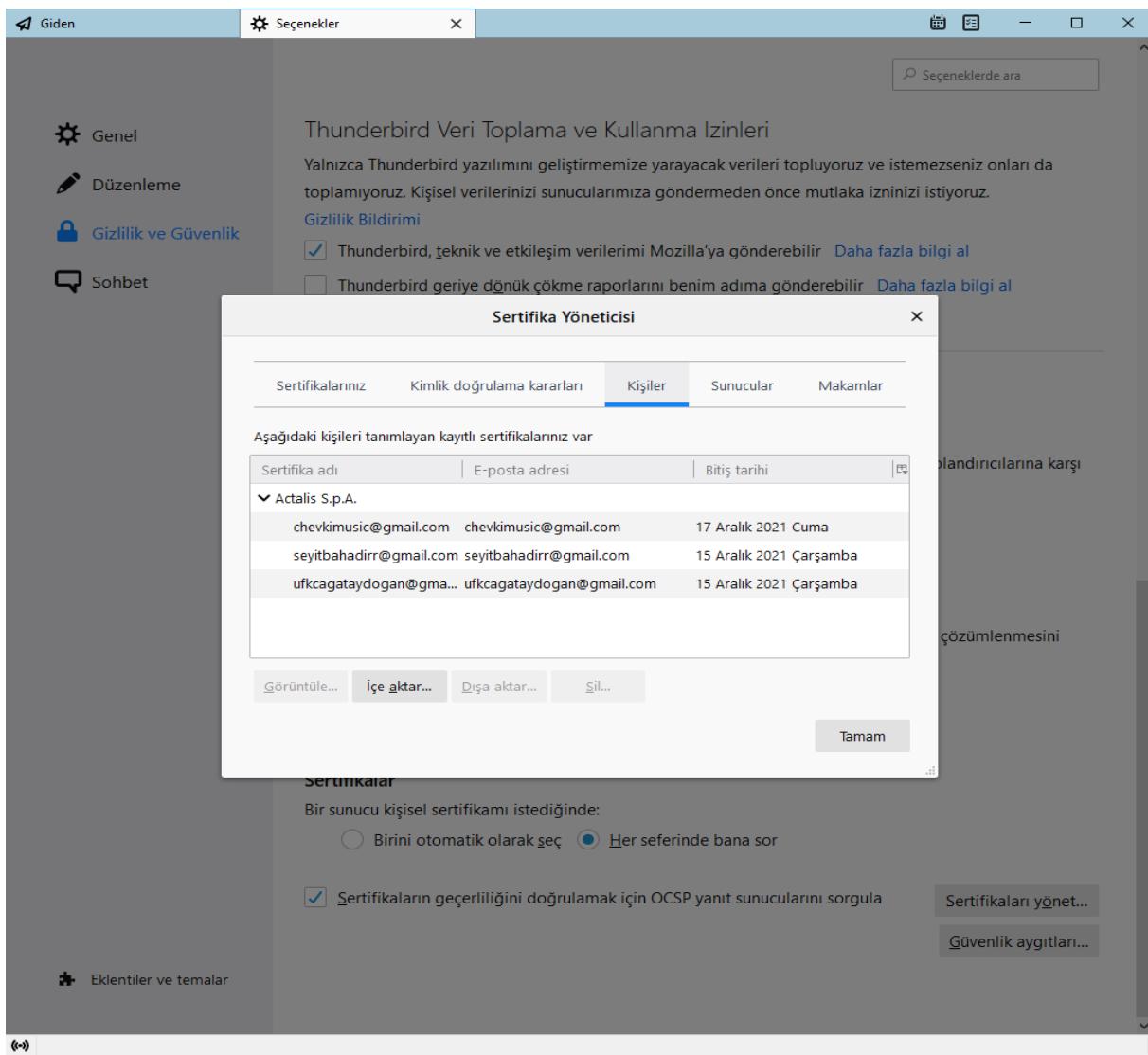
İleti şifrelenmiş
Bu ileti size gönderilmeden önce şifrelenmiş. Şifreleme, bu iletinin ağ üzerinde gezen başkaların tarafından okunmasını son derece zorlaştırır.

Okunmayan: 12 Toplam: 122

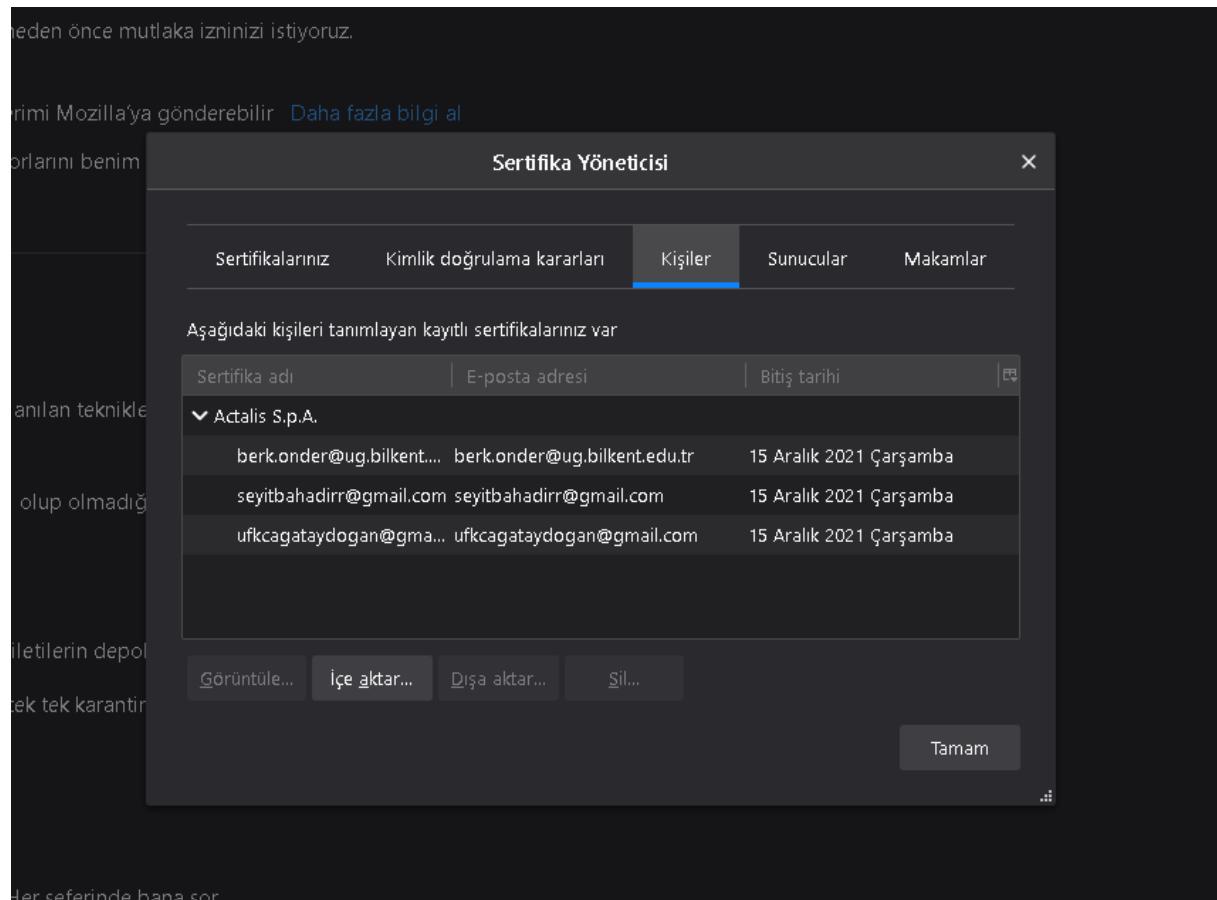
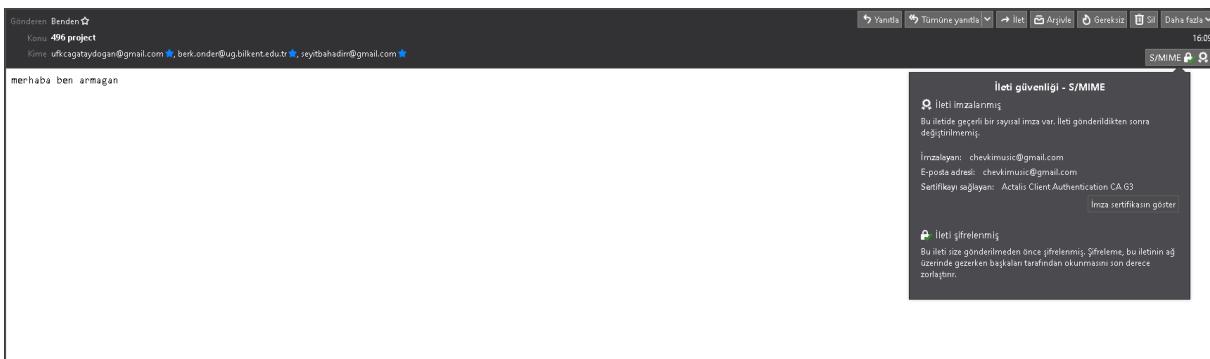
The screenshot shows the 'Sertifika Yöneticisi' (Certificate Manager) application window. The 'KİŞİLER' tab is selected. It displays a list of certificates for 'Actalis S.p.A.' with columns for 'Sertifika adı', 'E-posta adresi', and 'Bitiş tarihi'. Below the table are buttons for 'Görüntüle...', 'İçe aktar...', 'Dışa aktar...', 'Sil...', and 'Tamam'. At the bottom, there's an S/MIME encrypted email interface with fields for 'Gonderen', 'Konu', 'Kime', and a message body. A large callout box provides detailed information about the message's security, mentioning digital signatures and encryption. The bottom right corner shows statistics for unread and total messages.

Berk ÖNDER Screenshots:





Armağan OĞUZ Screenshots:



3. Part of The Project: Configuring Apache Webserver

Çağatay DOĞAN Screenshots:

The screenshot shows two terminal windows side-by-side, both titled "Linux Lite Terminal -". The top terminal window displays the process of unpacking various packages and installing Apache2. It shows errors related to firmware installation and dependency issues. The bottom terminal window shows the configuration of SSL/TLS for Apache, including setting up modules, generating RSA private keys, and reloading the service. Both terminals have a dark theme with a red feather icon in the background.

```
File Edit View Terminal Tabs Help
Unpacking intel-microcode (3.20201110.0ubuntu0.20.04.2) over (3.20200609.0ubuntu0.20.04.2) ...
Preparing to unpack .../69-zfs-initramfs_0.8.3-1ubuntu12.5_amd64.deb ...
Unpacking zfs-initramfs (0.8.3-1ubuntu12.5) over (0.8.3-1ubuntu12.4) ...
Preparing to unpack .../70-zfsutils-linux_0.8.3-1ubuntu12.5_amd64.deb ...
Unpacking zfsutils (0.8.3-1ubuntu12.5) over (0.8.3-1ubuntu12.4) ...
Preparing to unpack .../71-libutil1linux_0.8.3-1ubuntu12.5_amd64.deb ...
Unpacking libutil1linux (0.8.3-1ubuntu12.5) over (0.8.3-1ubuntu12.4) ...
Preparing to unpack .../72-libzfs2linux_0.8.3-1ubuntu12.5_amd64.deb ...
Unpacking libzfs2linux (0.8.3-1ubuntu12.5) over (0.8.3-1ubuntu12.4) ...
Preparing to unpack .../73-libzpool2linux_0.8.3-1ubuntu12.5_amd64.deb ...
Unpacking libzpool2linux (0.8.3-1ubuntu12.5) over (0.8.3-1ubuntu12.4) ...
Preparing to unpack .../74-libnvpair1linux_0.8.3-1ubuntu12.5_amd64.deb ...
Unpacking libnvpair1linux (0.8.3-1ubuntu12.5) over (0.8.3-1ubuntu12.4) ...
Preparing to unpack .../75-zfs-zed_0.8.3-1ubuntu12.5_amd64.deb ...
Unpacking zfs-zed (0.8.3-1ubuntu12.5) over (0.8.3-1ubuntu12.4) ...
Errors were encountered while processing:
/ttmp/apt-dpkg-install-dBXAYy/49-linux-firmware_1.187.6_all.deb
E: Sub-process /usr/bin/dpkg returned an error code (1)
linux ~ 100 sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
apache2-bin apache2-data apache2-utils libapr1 libaprutil1
libaprutil1-dbd-sqlite3 libaprutil1-ldap
Suggested packages:
apache2-doc apache2-suexec-pristine | apache2-suexec-custom ufw
The following NEW packages will be installed:
apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 upgraded, 8 newly installed, 0 to remove and 4 not upgraded.
103 not fully installed or removed.
Need to get 1,713 kB of archives.
After this operation, 7,494 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 libapr1 amd64 1:6.5-1ubuntu1 [914 kB]
Menu 🔍 🌐 📁 🗃 🔍 Linux Lite Terminal -
```



```
File Edit View Terminal Tabs Help
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for initramfs-tools (0.136ubuntu6.3) ...
update-initramfs is disabled since running on read-only media
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libglib2.0-0:amd64 (2.64.3-1~ubuntu20.04.1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.1) ...
Processing triggers for systemd (245.4-4ubuntu3.3) ...
Processing triggers for gconf2 (3.2.6-6ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libgdk-pixbuf2.0-0:amd64 (2.40.0+dfsg-3ubuntu0.1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for shared-mime-info (1.15-1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
linux ~ sudo a2enmod ssl
Considering dependency setenovif for ssl:
Module setenovif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
systemctl restart apache2
linux ~ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
systemctl reload apache2
linux ~ sudo service apache2 reload
linux ~ sudo mkdir /etc/apache2/ssl
linux ~ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.e.crt
Generating a RSA private key
+++++
Menu 🔍 🌐 📁 🗃 🔍 Linux Lite Terminal -
```

Linux Lite Terminal -

```

File Edit View Terminal Tabs Help
Email Address []:ufkcagataydogan@gmail.com
linux ~ sudo chmod 600 /etc/apache2/ssl/*
linux ~ sudo nano /etc/apache2/sites-enabled/default-ssl.conf
linux ~ sudo service apache2 reload
linux ~ hostname -I
10.0.2.15
linux ~ openssl s_client -connect 10.0.2.15:443
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 C = TR, ST = Ankara, L = Yenimahalle, O = Bilkent University, OU = SSL Certificate, CN = example.com, emailAddress = ufkcagataydogan@gmail.com
verify error:num=18:self signed certificate
verify return:1
depth=0 C = TR, ST = Ankara, L = Yenimahalle, O = Bilkent University, OU = SSL Certificate, CN = example.com, emailAddress = ufkcagataydogan@gmail.com
verify return:1
---
Certificate chain
  0 s:C = TR, ST = Ankara, L = Yenimahalle, O = Bilkent University, OU = SSL Certificate, CN = example.com, emailAddress = ufkcagataydogan@gmail.com
    i:C = TR, ST = Ankara, L = Yenimahalle, O = Bilkent University, OU = SSL Certificate, CN = example.com, emailAddress = ufkcagataydogan@gmail.com
  ---
Server certificate
-----BEGIN CERTIFICATE-----
MIETOTCAgAwIBAgIUUQYmkKcY03t1GR0G8Y+EslJvqYwwDQYJKoZTlhcNAQEL
BQAwgaxCzAJBgNVBAYTAjRSMDQYDVQQIDAZBbmthcmExFDASBgNVBAcMC1I
bmtYwhhGxIMRsQwDQYDVQQDBCaWxrZW50IFVuaxZlcnNpdHkxDGAWbgNVBAsm
D1NTTCBDZXJ0aWzpYZFOZTEUMBIGA1UEAwLZXhhbXBsZS5jb20wXDAmBgkqhkiG
9wOBCQEWGXVma2NhZ2ZFOYX1kb2dhbkBnbWFpbC5jb20wHhcNMjAxMTE2MTQzMzUy
WhcNMjexMjE2MTQzMzUykJCBqZELMAkGA1UEBhMCVFlxDzANBgNVBAgMBkFua2Fy
YTEUMBIGA1UEBwwLwVuawIhaGFsbGUxGzAzBgNVBAoMEkJpbG1bnQgVW5pdMVy
c210deTEYMBGATUECwPUTNMIEn1cnpRznljYXR1MRQwEgYDVQDAtleGFtcGxI
LmNvbTEoMCYGSqSGS1b3DQEJARYZdwZrY2FnYXReWRvZ2FuQGdtYWlsLmNvbTCC
ASIwDQYJKoZIhvcNAQEBBQAdggEPADCCAQggEBANLV1/qAAHiKTWhfZtrohjC
RIRiTxD50d1Fp0xvV+RRF1C3eoA187mWh172vWl1IrnkWRvv6GT+FvvrYpW2rWk
-----END CERTIFICATE-----
Menu 🌐 🌐 🌐 🌐 🌐 Linux Lite Terminal - 8:26:18 am
```

Linux Lite Terminal -

```

File Edit View Terminal Tabs Help
linux ~ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
Generating a RSA private key
....+++++
.....+++++
writing new private key to '/etc/apache2/ssl/apache.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) [AU]:TR
State or Province Name (full name) [Some-State]:Ankara
Locality Name (eg, city) []:Yenimahalle
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Bilkent University
Organizational Unit Name (eg, section) []:SSL Certificate
Common Name (e.g. server FQDN or YOUR name) []:example.com
Email Address []:ufkcagataydogan@gmail.com
linux ~ sudo chmod 600 /etc/apache2/ssl/*
linux ~ sudo nano /etc/apache2/sites-enabled/default-ssl.conf
linux ~ sudo service apache2 reload
linux ~ hostname -I
10.0.2.15
linux ~ openssl s_client -connect 10.0.2.15:443
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 C = TR, ST = Ankara, L = Yenimahalle, O = Bilkent University, OU = SSL Certificate, CN = example.com, emailAddress = ufkcagataydogan@gmail.com
verify error:num=18:self signed certificate
verify return:1
depth=0 C = TR, ST = Ankara, L = Yenimahalle, O = Bilkent University, OU = SSL Certificate, CN = example.com, emailAddress = ufkcagataydogan@gmail.com
verify return:1
-----
Menu 🌐 🌐 🌐 🌐 🌐 Linux Lite Terminal - 8:26:09 am
```

```

Linux Lite Terminal -
File Edit View Terminal Tabs Help
i:C = TR, ST = Ankara, L = Yenimahalle, O = Bilkent University, OU = SSL Certificate, CN = example.com, emailAddress = ufkcagataydogan@gmail.com
-- 
Server certificate
-----BEGIN CERTIFICATE-----
MIIEOTCAyGgAwIBAgIUVQYmkKcYO3t1GR0G8Y+EslJvqYwvDQYJKoZIhvcNAQEL
BQAwgasxCzAJBgNVBAYTAlRSM08wDQYDVQQIDAZBbmthcmExFDASBgNVBACMC1ll
bmltYwhhbGxIMRswG0YDVQQKDBJCwtxrZW50fVuaXz1cnNpdHkxDWBgNVBAsM
D1NTTCBDZXJ0aWzjP2F0ZTEUMBIGA1UEAwLZXhhbBsZS5jb20wKDAmbGkgkhkIG
9wNCMMjeXmjE2MT0xM2UyWjCBzELMAkGA1UEBhMCVFIxDzANBgNVBAgMBkFua2Fy
YTEUMBIGA1UEBwwLWwVuaWhaGfsGUxGzBgvNBAAoMEJpbGtlnQgvW5pdmVy
c210eTEYMBYGA1UECwwPU1NMIE1cnRpZmljYXR1MRQwEgYDVQODAtleGFtcGxI
LmNvbT0oMCYGSgS1b3DQEJARYZDZrY2fNXXRheWRvZzFuQGdtYwlsNmNvbTCC
ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQcggEBANLV1/qAAHiKTWhfZtrohjG
8JBUTxh50d1feoxyv+B8F1C3eoA18zmWu72CyWJJrokWBvv6GT+FvvrYpWv8wK
0YCGkaIMtOKXlUFMwmK2qm0zofqAwleWRcofvxEcpZ3e8ejxgDQpV77msTQgdD
07JJoah12E5eMjtws6f+hh87R6CmzxZfREMqu14u4nP0tqBL009IyEt17q4aPA
ttl8VQPipiu0SmvIyhanh1mN/U64rKLtyrFULGJTJFjrv9RHyQFG3eiGHh/9q2xt
39zmk3dk3akVR0NKTcwgvUe1NouxvG2DcvKESkpj/91+Yxm+/H1a+0yrn/gSVsC
AwEAAaNTMFewH0QYDVR0OBYEFMorzj210KA23VPfS6kt4DgZ2+g6MB8GA1UdIwOY
MBaAFMorzj210KA23VPfS6kt4DgZ2+g6MA8GA1UdEwEB/wQFMAmAf8wDQYJKoZI
hvcNAQELBQADggEBAM1TuyXLjmSYp+ZK2r/MoRmGi0aquWXWTOEwrtpxJx6T48Sz+
nAbvP3mCOVOnzP3qtaHDRTUPX9+MmNg2z6BExC07IKu09haV2h39Pp/+LgDguu
ibG3K/rCe87hDwJGhmeed4b+mSmb+ZzePLZSiINJKGUzyC3rQOWFpOEAxhdhDyf
Qn7cbgqjo/IaB6TFP2F62ua7Ciyses5Jvd3hZiBw9edr4fsd0jwyFA40w8wldDi
MpR94LKdWjy0ZtCvS/bCEXaxoSohaNMbMk50DrSm2Ataz108wuiTSPL1m/Aw/BSH
RBJKJLhc31f6x76zhbNghKf1Nwqn5foPxMs8g=
-----END CERTIFICATE-----
subject=C = TR, ST = Ankara, L = Yenimahalle, O = Bilkent University, OU = SSL Certificate, CN = example.com, emailAddress = ufkcagataydogan@gmail.com
issuer=C = TR, ST = Ankara, L = Yenimahalle, O = Bilkent University, OU = SSL Certificate, CN = example.com, emailAddress = ufkcagataydogan@gmail.com
Linux Lite Terminal -
File Edit View Terminal Tabs Help
-----END CERTIFICATE-----
subject=C = TR, ST = Ankara, L = Yenimahalle, O = Bilkent University, OU = SSL Certificate, CN = example.com, emailAddress = ufkcagataydogan@gmail.com
issuer=C = TR, ST = Ankara, L = Yenimahalle, O = Bilkent University, OU = SSL Certificate, CN = example.com, emailAddress = ufkcagataydogan@gmail.com
Linux Lite Terminal -
File Edit View Terminal Tabs Help
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
...
SSL handshake has read 1641 bytes and written 363 bytes
Verification error: self signed certificate
...
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 18 (self signed certificate)
...
Post-Handshake New Session Ticket arrived:
SSL-Session:
  Protocol : TLSv1.3
  Cipher   : TLS_AES_256_GCM_SHA384
  Session-ID: 60E57FE2E174C3513DDF4C4ADA83B319AE02B59E2A18DBFE8DD390ED4AF7CFCD
  Session-ID-cxt:
  Resumption PSK: 1DFBD8D2325DB367EE43BAF1AA2144472B30F5B09D97FCF01A540076ACD8FBDB9F032126304E7AAFB758C85051BF884F
  PSK identity: None
  PSK identity hint: None
  SRP username: None
Linux Lite Terminal -

```

Bahadır ÇETİN Screenshots:

```
kali㉿kali:~$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.43-1).
apache2 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
kali㉿kali:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
kali㉿kali:~$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
    systemctl reload apache2
kali㉿kali:~$ hostname -I
192.168.177.129
kali㉿kali:~$ openssl s_client -connect 192.168.177.129:443
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 C = TR, ST = Cankaya, L = Ankara, O = Bahadir Cetin, OU = Section 1, CN = example.com, emailAddress = seyitbahadir@gmail.com
verify error:num=18:self signed certificate
verify return:1
depth=0 C = TR, ST = Cankaya, L = Ankara, O = Bahadir Cetin, OU = Section 1, CN = example.com, emailAddress = seyitbahadir@gmail.com
verify return:1
---
Certificate chain
  0 s:C = TR, ST = Cankaya, L = Ankara, O = Bahadir Cetin, OU = Section 1, CN = example.com, emailAddress = seyitbahadir@gmail.com
    i:C = TR, ST = Cankaya, L = Ankara, O = Bahadir Cetin, OU = Section 1, CN = example.com, emailAddress = seyitbahadir@gmail.com
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIEFzCCAv+gAwIBAgIUK9JpdKuPwhODrlBpiuPjig8qbswDQYJKoZIhvcNAQEL
BQAwgZoxCzAJBgNVBAYTAlRSMRAwDgYDVQQIDAoDYW5rYXlhMQ8wDQYDVQQHDAZB
omthcmExFjAUBgNVBAoMDUJhaGFkaXIgQ2V0aW4xEjAQBgNVBAAsMCVNLY3Rpb24g
MTEUMBIGA1UEAwWLZhbbXBsZS5jb20xJjAkBgkqhkiG9w0BCQEWF3NleWl0YmFo
YWRpcnJAZ21haWwUY29tMB4XDITiwMTIxNjExNTMzM1oXDITxMTIxNjExNTMzM1ow
gZoxCzAJBgNVBAYTAlRSMRAwDgYDVQQIDAoDYW5rYXlhMQ8wDQYDVQQHDAZBbmth
cmExFjAUBgNVBAoMDUJhaGFkaXIgQ2V0aW4xEjAQBgNVBAAsMCVNLY3Rpb24gMTEU
MBIGA1UEAwWLZhbbXBsZS5jb20xJjAkBgkqhkiG9w0BAQEFAAOCAQ8AMIIIBCgKCAQE
AqOBbL4xZlMzfx7PNVWaIUm4Se3sXTIdF13T+0Ucnp9xmb6kGAVnQzcX0iawsg7BeBHQ
D6hNbMLDjjER0u3ay84jDyIae6AAhA4puJeUliua7JokXm3r+clpDLSWc62sjh0
dHFf9ZpF0sYSithDMc/BghsKS/ItsyGlJrK/e+K5scvifHK9z5Jejqeam6k+hwNq
e9Vr+j0nzmGbT9N6+6bQPbDhqM8UFsleCd0k7zAeiU6FCXV/TbXRWEdkKexflr3/
NSxjfvc8I3WjUtXSx2tmdb85KC9JJD6PFfe6ky935aaajwpUxTuIIH9iesD3DeoCx
aLW/LKxt0hnF7LM14AwIDAQABo1MwUTAdBgnVHQ4EfgQuNteSlEyJXndLATfJT4FJ
g3aD/mQwHwYDVR0jBBgwFoAUuTeSlEyJXndlATfJT4FJg3aD/mQwDwYDVR0TAQH/
BAUwAwEB/zANBgkqhkiG9w0BAQsFAAACQEAAMAaoh9VXloNvLwmcBUbjE2dHu6/
```

```
kali㉿kali:~$ sudo service apache2 start
kali㉿kali:~$ sudo service apache2 reload
kali㉿kali:~$ sudo mkdir /etc/apache2/ssl
kali㉿kali:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
Generating a RSA private key
-----+
writing new private key to '/etc/apache2/ssl/apache.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) [AU]:TR
State or Province Name (full name) [Some-State]:Cankaya
Locality Name (eg, city) []:Ankara
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Bahadir Cetin
Organizational Unit Name (eg, section) []:Section 1
Common Name (e.g. server FQDN or YOUR name) []:example.com
Email Address []:seyitbahadirc@gmail.com
kali㉿kali:~$ sudo chmod 600 /etc/apache2/ssl/*
kali㉿kali:~$ sudo nano /etc/apache2/sites-enabled/default-ssl.conf
kali㉿kali:~$ sudo service apache2 reload
kali㉿kali:~$ openssl s_client -connect 192.168.17.129:443
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 C = TR, ST = Cankaya, L = Ankara, O = Bahadir Cetin, OU = Section 1, CN = example.com, emailAddress
= seyitbahadirc@gmail.com
verify error:num=18:self signed certificate
verify return:1
depth=0 C = TR, ST = Cankaya, L = Ankara, O = Bahadir Cetin, OU = Section 1, CN = example.com, emailAddress
= seyitbahadirc@gmail.com
verify return:1
-----
Certificate chain
 0 s:C = TR, ST = Cankaya, L = Ankara, O = Bahadir Cetin, OU = Section 1, CN = example.com, emailAddress =
seyitbahadirc@gmail.com
  i:C = TR, ST = Cankaya, L = Ankara, O = Bahadir Cetin, OU = Section 1, CN = example.com, emailAddress =
seyitbahadirc@gmail.com
-----
Server certificate
-----BEGIN CERTIFICATE-----MIIEFzCCAvwIBAgUK9JpkUlwPh0DrhlBpijPjig8qbwsuQYKjOZIhvNAQEL
BWAwgZoxCzAjbgbNVAYTlRSMRAdgYDQVQIDAdWY5VY1hM0QbwQDYDQVQHDAZB
bmhcmExfjAubgNVBAoMDUjhAGfkxAxQ2vwAwxjxAQBgNVBAsMCVNL3RpB24g
MTUEUBTG1AUeAwLZXhbxBSzSSjb20xJjAkBgkqhkiG10w0BCQEWf3N1wL0YmFo
YWRpcnJA221haWu29tMBx4XTtIwNxNjExNTMzM1oXTIxMTIxNjExNTMzM1ow
gZoxCzAjbgbNVAYTlRSMRAdgYDQVQIDAdWY5VY1hM0QbwQDYDQVQHDAZBmt
cmExfjAubgNVBAoMDUjhAGfkxAxQ2vwAwxjxAQBgNVBAsMCVNL3RpB24gMTEU
MTBG1AUeAwLZXhbxBSzSSjb20xJjAkBgkqhkiG10w0BCQEWf3N1wL0YmFoYWRp
cJAZ221haWu29tMTIBTjANBgkqhkiG10w0BAQEAAQBM1TCgKCAQEAqBb
L4xZM2rfx7PNWlAIImSe3+XTTdfF13T+0Ucnrpxmb6GVNvQZcxiawsg7B8eHQ
D6hNMMLDjER0U3A/84JjDyIae6AhAp4uJeUliu5TjokXn3+r+clpLSMc62sjh0
dHFF92pF0sYsItihDm/BghsKS/ItsyGLjr/+K5scviFHk9Z5jeqean6k+hWnQ
j0nzmgbT9M6-6oQpDhM0BFUsflecd0k/zAe1U6FCXV/TbKRWedKexfr3/
NSjx+jU8bgVBAoMDUjhAGfkxAxQ2vwAwxjxAQBgNVBAsMCVNL3RpB24gMTEU
LW/Lkxt0hnf7LM4jUtwIAb01MwJTAtdBgVNWQ4EfgQuTnsLeyJXndlATFjT4Fj
g3dAmQhWhDVROjBggwFoAUtSLeYJXndlATFjT4Fjg3dAmQhWhDVROtAQH
BALUAbwEB/zANBgkqhkiG10w0BAQEFAAQEAAMAAoh9V1xLNwLumbcblje2Hu6w
/Y20AcxKvkTxWf0jCB3ctFv587Czy2Uk4fNnSAIMakd8LSDU0f016wenghzrly. You should replace this file (located at /var/www/
kwad9qVGrifcfwzg5t8Z3G+OymNndgyc2UjNrrYVd4hn7zkqNU
D71W04LabhvrM3y/fgiX0UWHAs5J28jiYTURd2pEwR0/PoApH4c4Qag=
-----END CERTIFICATE-----subject=C = TR, ST = Cankaya, L = Ankara, O = Bahadir Cetin, OU = Section 1, CN = example.com, emailAddress
= seyitbahadirc@gmail.com
-----openssl Anaconda configuration-----The Apache2 configuration is different from the upstream version. The configuration, and split into several files for interaction with Debian tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation.
Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Debian systems is as follows:
-----No client certificate CA names sent-----Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
-----
SSL handshake has read 1607 bytes and written 363 bytes
Verification error: self signed certificate
-----New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 18 (self signed certificate)
-----
-----Post-Handshake New Session Ticket arrived:
SSLSession:
  Protocol : TLSv1.3
  Cipher : TLS_AES_256_GCM_SHA384
  Session-ID: F273F408753CAB15DEE120D7B6D400F940DE89F13C9C2C5CE31802E6AA5FF92
  Session-ID-ctx:
  Resumption PSK: F17B114901BD309A8D525CD5186E3811B8A666F37D8468DC778C2E662F3DA561D621F6D09DB460FDA12352
D2EBB86B7D
  PSK identity: None
  PSK identity hint: None
```

Berk ÖNDER Screenshots:

```
issuer=C = TR, ST = Ankara, L = Cankaya, O = Bilkent, OU = CTIS, CN = example.com, emailA
der@ug.bilkent.edu.tr

---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 1595 bytes and written 363 bytes
Verification error: self signed certificate
---
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 18 (self signed certificate)
---
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol : TLSv1.3
    Cipher   : TLS_AES_256_GCM_SHA384
    Session-ID: 04EBDCE869A87DB9C67127921FE00E30AC405AE6A7E82F4F249DB0C7CB2EA72B
    Session-ID-ctx:
linux ~ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
linux ~ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
    systemctl reload apache2
linux ~ systemctl restart apache2
linux ~ systemctl reload apache2
linux ~ sudo mkdir /etc/apache2/ssl
linux ~ setxkbmap tr
linux ~ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key
-out /etc/apache2/ssl/apache.crt
Generating a RSA private key
```

```
writing new private key to '/etc/apache2/ssl/apache.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) [AU]:TR
State or Province Name (full name) [Some-State]:Ankara
Locality Name (eg, city) []:Cankaya
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Bilkent
Organizational Unit Name (eg, section) []:CTIS
Common Name (e.g. server FQDN or YOUR name) []:example.com
Email Address []:berk.onder@ug.bilkent.edu.tr
linux ~ sudo chmod 600 /etc/apache2/ssl/*
linux ~ sudo nano /etc/apache2/sites-enabled/default-ssl.conf
linux ~ sudo service apache2 reload
linux ~ hostname I
hostname: you must be root to change the host name
linux ~ 1 hostname -I
10.0.2.15
linux ~ openssl s_client -connect 10.0.2.15:443
CONNECTED(00000003)
```

Armağan OĞUZ Screenshots:

Linux Lite Terminal -

```

Dosya Düzenle Görünüm Uçbirim Sekmeler Yardım
Post-Handshake New Session Ticket arrived:
SSL-Session:
Protocol : TLSv1.3
Cipher  : TLS_AES_256_GCM_SHA384
Session-ID: BFBAF5B7059B414BE0186891EE7D13A50A12137F445D8315F2B16E7EDF196C4E
Session-ID-ctx:
Resumption PSK: C97098619C5337A6133BCEE049979C1DBDD59AC81B8B28E11E5DBCBA71DA
AAE466941E7DF03FC79A8ECA8EAD1EB6A92
PSK identity: None
PSK identity hint: None
SRP username: None
TLS session ticket lifetime hint: 300 (seconds)
TLS session ticket:
0000 - 2b 3f 9f fc f8 d1 37 cb-97 43 77 e5 2d a8 46 59 +?....7..Cw.-.FY
0010 - 64 b2 45 c4 e0 bb fc bc-68 2c bf ec 57 23 9a a1 d.E,...h...W#..
0020 - 0d b6 c5 bd f8 d0 88 df-25 6b 83 04 3e fb 4f 49 .....%K..>.OI
0030 - 6f 48 78 ce b3 3c 32 c9-0f f1 65 70 3d 4a a2 3e oHx..<...ep=J.>
0040 - 75 61 61 db 33 42 90 3c-08 03 4d 5c 5b 0e 3d dd uaa.3B.<..M\|.=.
0050 - e4 a2 27 67 11 3a 02 02-5e a5 2f 87 15 3b d7 13 ..'g...^/.....
0060 - 1b a8 a3 a3 9d ec 67 b3-57 e1 6e 38 e0 a0 3f 1f .....g.W.n8.?
0070 - f1 10 39 33 75 44 b6 69-8f 08 f5 83 2e e9 27 9e ..93uD.i....'.
0080 - 1e 92 b0 6e fd 71 0f a5-66 f3 33 f3 9a 02 bf a1 ...n.q.f.3....
0090 - f6 d2 b8 53 a3 67 ee 02-97 c0 3b 09 8c 3b d3 38 ...S.g....;.;8
00a0 - db 3f 3b 5c bb f4 10 fc-60 7d 71 47 9c 6c ad a1 ..;\....)qG.l..
00b0 - b7 f1 36 c6 76 42 e0 f6-2e 8d df 80 2d 30 89 28 ..6.vB....-0.(.
00c0 - a2 f8 e2 5d c5 d8 00 ad-32 50 8f 0b 01 74 dd d0 ..]....2P...t..
00d0 - ff 43 d2 f8 e1 5b 74 34-bd 0e cf 5f e9 c9 7a c6 .C...[t4....z.

Start Time: 1608213143
Timeout   : 7200 (sec)
Verify return code: 18 (self signed certificate)
Extended master secret: no
Max Early Data: 0
...
read R BLOCK
closed
khrystal06 ~ ^C
khrystal06 ~ 130 ^C
khrystal06 ~ 130

```

Linux Lite Terminal -

```

Dosya Düzenle Görünüm Uçbirim Sekmeler Yardım
Tetikleyiciler işleniyor: man-db (2.9.1-1) ...
Tetikleyiciler işleniyor: libc-bin (2.31-0ubuntu9.1) ...
khrystal06 ~ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
khrystal06 ~ systemctl restart apache2
khrystal06 ~ sudo a2ensite default-ssl
^C
khrystal06 ~ 130 ~ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
    systemctl reload apache2
khrystal06 ~ systemctl reload apache2
khrystal06 ~ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
khrystal06 ~ sudo a2ensite default-ssl
Site default-ssl already enabled
khrystal06 ~ sudo service apache2 reload
khrystal06 ~ sudo mkdir /etc/apache2/ssl
khrystal06 ~ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
Generating a RSA private key
.....+++++
.....++++

```

Linux Lite Terminal -

Dosya Düzenle Görünüm Uçbirim Sekmeler Yardım

Organizational Unit Name (eg, section) []:SSL Certificate Test
Common Name (e.g. server FQDN or YOUR name) []:armagan

```
khrystal06 ~ sudo chmod 600 /etc/apache2/ssl/*
khrystal06 ~ sudo nano /etc/apache2/sites-enabled/default-ssl.conf
khrystal06 ~ sudo service apache2 reload
khrystal06 ~ hostname -l
hostname: invalid option -- 'l'
Usage: hostname [-b] <hostname>-F file)      set host name (from file)
        hostname [-a|-A|-d|-f|-i|-I|-s|-y]    display formatted name
        hostname                                         display host name

        {yp,nis,}domainname {nisdomain|-F file}    set NIS domain name (from file)
        {yp,nis,}domainname                         display NIS domain name

        dnsdomainname                                display dns domain name

        hostname -V|--version|-h|--help            print info and exit

Program name:
        {yp,nis,}domainname=hostname -y
        dnsdomainname=hostname -d

Program options:
        -a, --alias          alias names
        -A, --all-fqdns      all long host names (FQDNs)
        -b, --boot           set default hostname if none available
        -d, --domain         DNS domain name
        -f, --fqdn, --long   long host name (FQDN)
        -F, --file           read host name or NIS domain name from given file
        -i, --ip-address     addresses for the host name
        -I, --all-ip-addresses all addresses for the host
        -s, --short          short host name
        -y, --yp, --nis       NIS/YP domain name

Description:
This command can get or set the host name or the NIS domain name. You can
also get the DNS domain or the FQDN (fully qualified domain name).
Unless you are using bind or NIS for host lookups you can change the
```

Linux Lite Terminal -

also get the DNS domain or the FQDN (fully qualified domain name).
Unless you are using bind or NIS for host lookups you can change the
FQDN (Fully Qualified Domain Name) and the DNS domain name (which is
part of the FQDN) in the /etc/hosts file.

```
khrystal06 ~ 255 > hostname -i
127.0.1.1
khrystal06 ~ openssl s_client -connect 127.0.1.1:443
CONNECTED(00000003)
Can't use SSL_get_servername
depth=0 C = TR, ST = Ankara, L = ANK, O = Armagan, OU = SSL Certificate Test, CN
= armagan, emailAddress = chevkimusic@gmail.com
verify error:num=18:self signed certificate
verify return:1
depth=0 C = TR, ST = Ankara, L = ANK, O = Armagan, OU = SSL Certificate Test, CN
= armagan, emailAddress = chevkimusic@gmail.com
verify return:1
---
Certificate chain
  0 s:C = TR, ST = Ankara, L = ANK, O = Armagan, OU = SSL Certificate Test, CN =
armagan, emailAddress = chevkimusic@gmail.com
      i:C = TR, ST = Ankara, L = ANK, O = Armagan, OU = SSL Certificate Test, CN =
armagan, emailAddress = chevkimusic@gmail.com
  ---
Server certificate
-----BEGIN CERTIFICATE-----
MIIEDTCCAwBgAwIBAgIUCQf1Mr0m3Vjj7gj4Wkv1dt/9CpMwDQYKoZihvCNAQEL
BQAwgZUxCzAJBgNVBAYTA1RSQ8wDQYDVQKIDAZBbmthcmExDDAKBgNVBAcMAOFO
SzE0MA4G1UECgwHQXJtYwdhbjEdMBsGA1UECwwUU1NMIE1lcRpZmljYXRlIFRl
c3QxEDAOBgNVBAMMB2FybWFnYW4xJDAiBgkqhkiG9w0BCQEwFwNoZXZraW11c2lj
QGdtYwlsLmNvbTaeFw0yMDEyMTcxMzQ0NTdaFw0yMTEyMTcxMzQ0NTdaMIGVMQsw
CQYDVQQGEwJUUjEPMA0GA1UECAwGQw5rYXJhMQwwCgYDVQQHDANBTksxEDAOBgNV
BAoMB0FybWFnYW4xHTAbBgNVBAsMFNNTTCBDZXJ0awZpY2F0ZSBUXN0MRawDgYD
VQDDADhcm1hZ2FuMSQwIgYJKoZihvNAQkBhFvjaGV2a21tdXNpY0BnbwFpbCsj
b20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDz0e/tv+7AQr3Szkt
on17le441da061VotGLuwX1mx+d+0JEmw9GPaBlfXI4kL+3fJ15dyJ5fNNQuir
9nvUxftxaZh/GKICZ8yY/GxrTrIGFcxBVpxqbLkxL0wB0z+rp14Mtr5EiPi
ufRCa44aEwJXSRT1ywmR+odxjl+uovNhkQl+18JbK0kZAr1Eb2TdC9avbtGPBhVD
HB/YDjGYMOEtX8swbaHdZwgNoH09Ti0WmrhsRwN4dNs5jceg7205PFnFrwOSv2o
L9ufDynZu0R3m7jIqb6cY9Q4GIovpLEo7WQw8134Q1qIET9Uv3X2oGLzu0C0vYr8
```

Linux Lite Terminal -

```
Linux Lite Terminal - _ □ ×

Dosya Düzenle Görünüm Uçbirim Sekmeler Yardım

w+aoEON0teeZ0VdpNf5t12yunRr0rSyijeINNICEtcnGmhW5kp01vPSgSSg5r8JW
5PhlfyfYqJVwGZHDHkfNdMPuhwJJDzjZh/9v5vTH5bCgtrxbbv0VdZmVchgj2Ttz
7ygPLHU6r0ZVsEoh9Hu0pPWwmxENfHExGo7aVbBI/54WX4r0MVx9cW0spmuwjnfJ
QF4WvnPZ8zchXEpsZVSW+YPn1nx0Tx0u4D5g6rdHHESv
-----END CERTIFICATE-----
subject=C = TR, ST = Ankara, L = ANK, O = Armagan, OU = SSL Certificate Test, CN
= armagan, emailAddress = chevkimusic@gmail.com

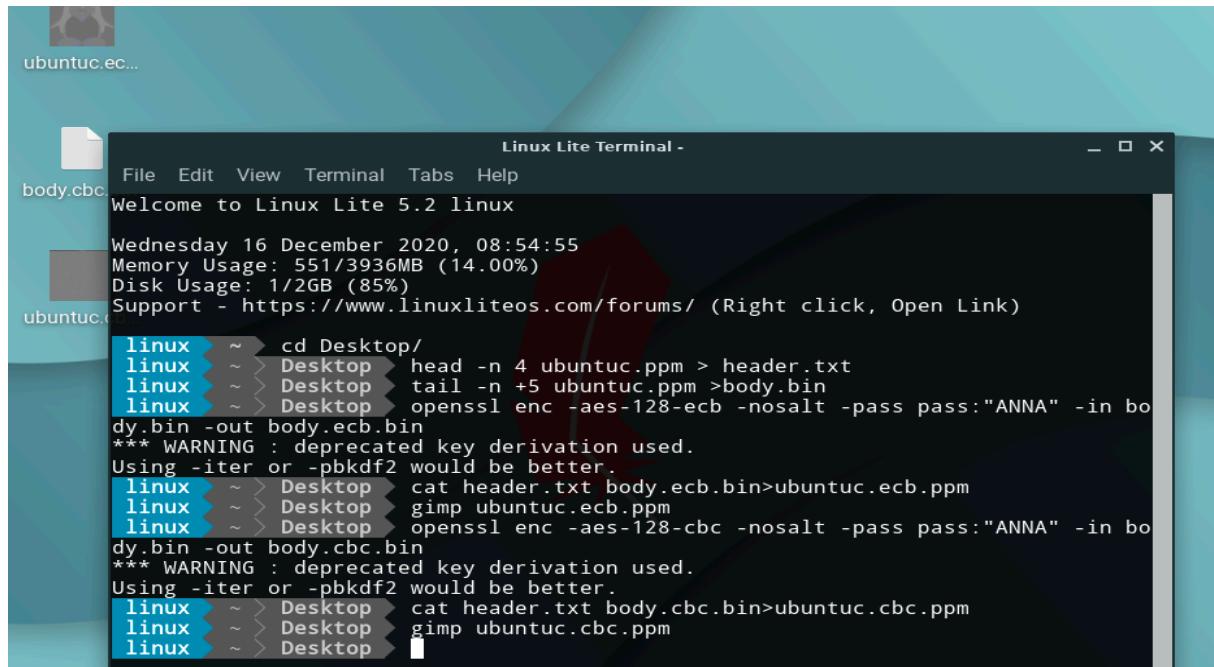
issuer=C = TR, ST = Ankara, L = ANK, O = Armagan, OU = SSL Certificate Test, CN
= armagan, emailAddress = chevkimusic@gmail.com

---
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 1597 bytes and written 363 bytes
Verification error: self signed certificate
---
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
```

4. Part of The Project: Encrypting an Image and Modifying AES.java

Encrypting an Image

Çağatay Screenshots:

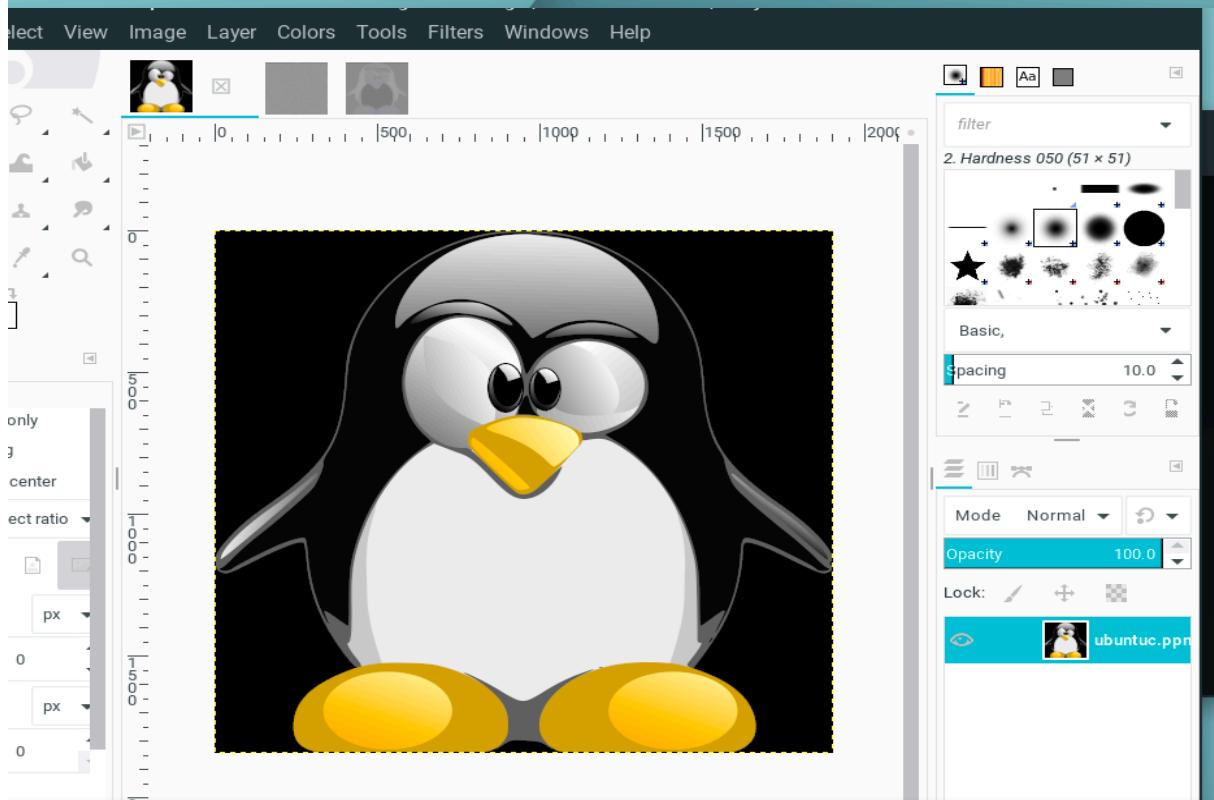


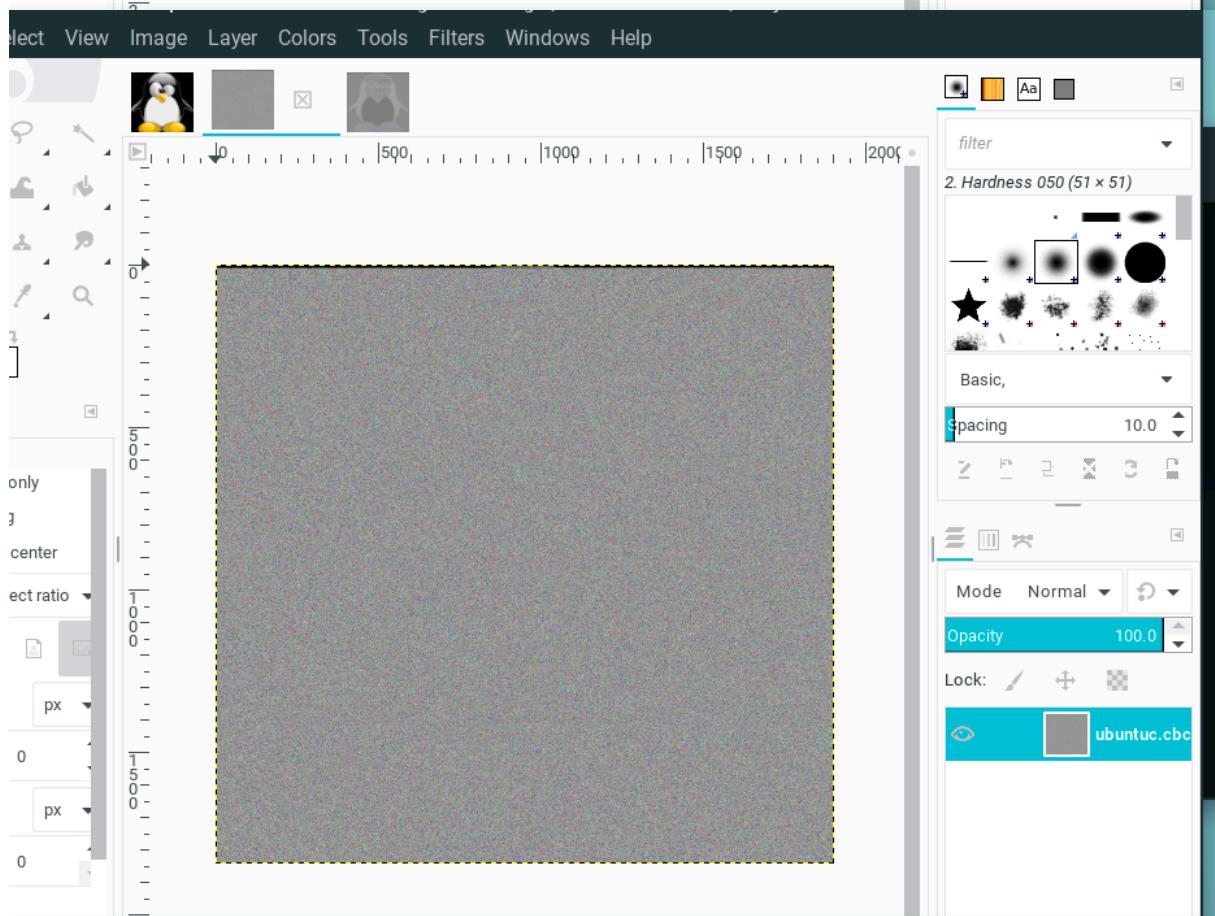
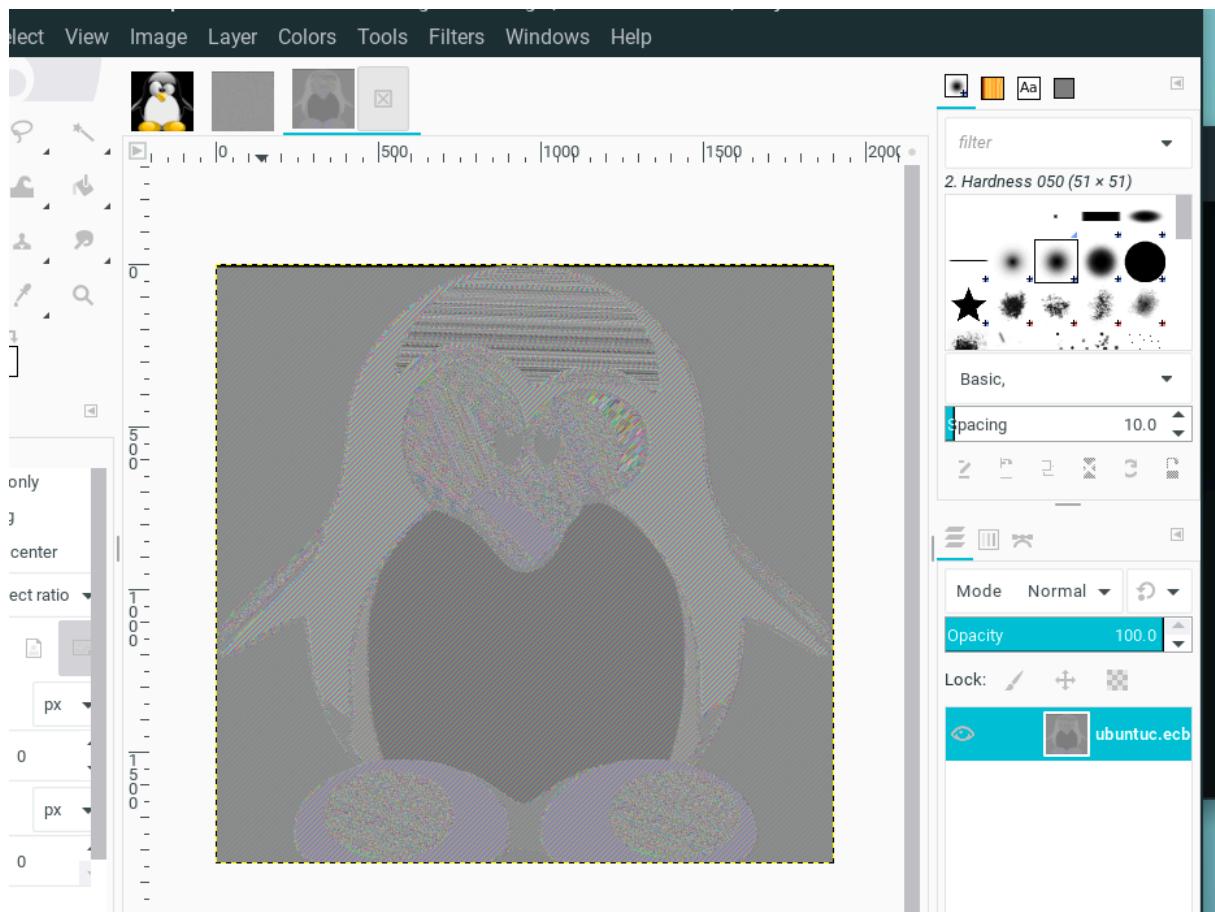
The screenshot shows a Linux Lite Terminal window with the following command history:

```
File Edit View Terminal Tabs Help
Welcome to Linux Lite 5.2 linux

Wednesday 16 December 2020, 08:54:55
Memory Usage: 551/3936MB (14.00%)
Disk Usage: 1/2GB (85%)
Support - https://www.linuxliteos.com/forums/ (Right click, Open Link)

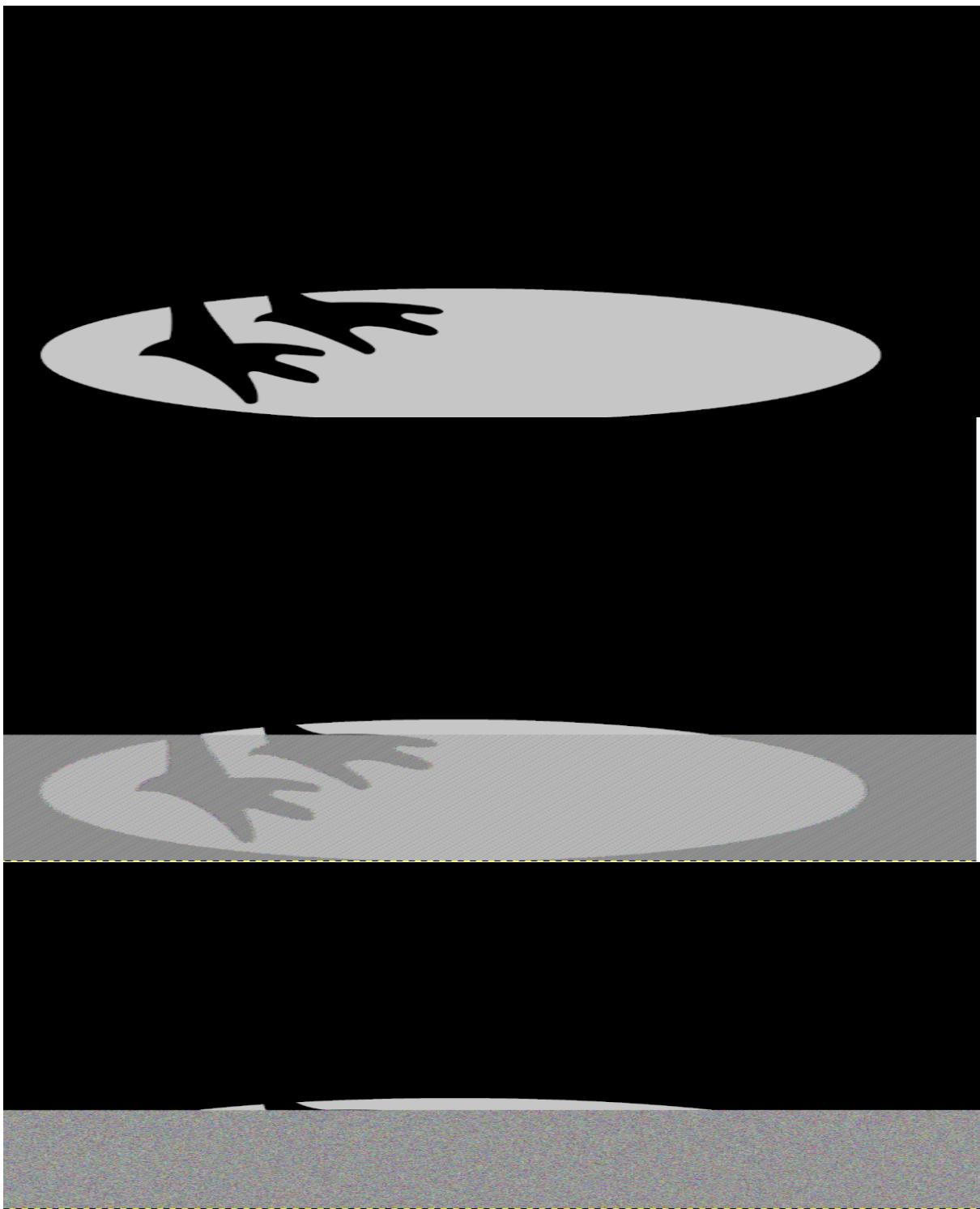
linux ~ > cd Desktop/
linux ~ > Desktop head -n 4 ubuntuc.ppm > header.txt
linux ~ > Desktop tail -n +5 ubuntuc.ppm >body.bin
linux ~ > Desktop openssl enc -aes-128-ecb -nosalt -pass pass:"ANNA" -in body.bin -out body.ecb.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
linux ~ > Desktop cat header.txt body.ecb.bin>ubuntuc.ecb.ppm
linux ~ > Desktop gimp ubuntuc.ecb.ppm
linux ~ > Desktop openssl enc -aes-128-cbc -nosalt -pass pass:"ANNA" -in body.bin -out body.cbc.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
linux ~ > Desktop cat header.txt body.cbc.bin>ubuntuc.cbc.ppm
linux ~ > Desktop gimp ubuntuc.cbc.ppm
linux ~ > Desktop
```





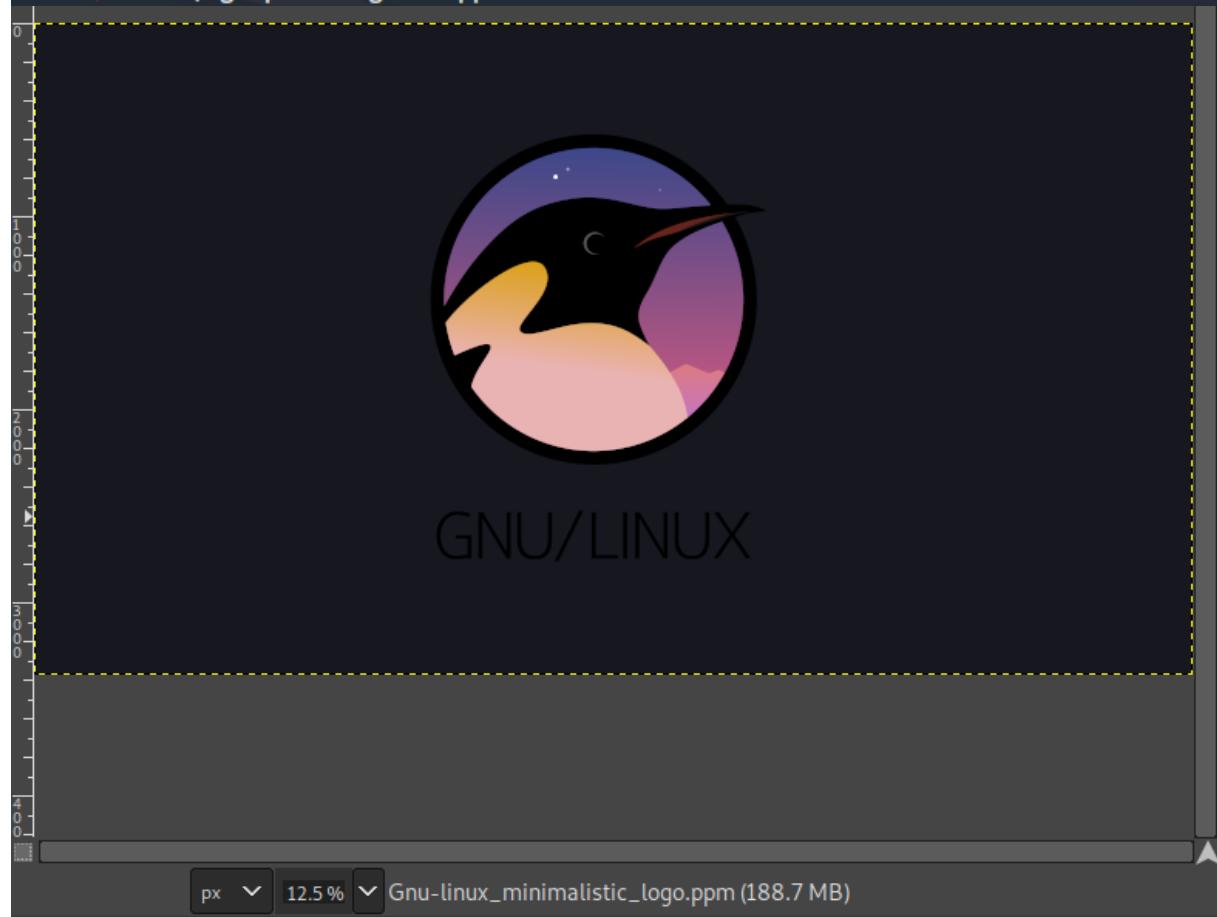
Berk ÖNDER Screenshots:

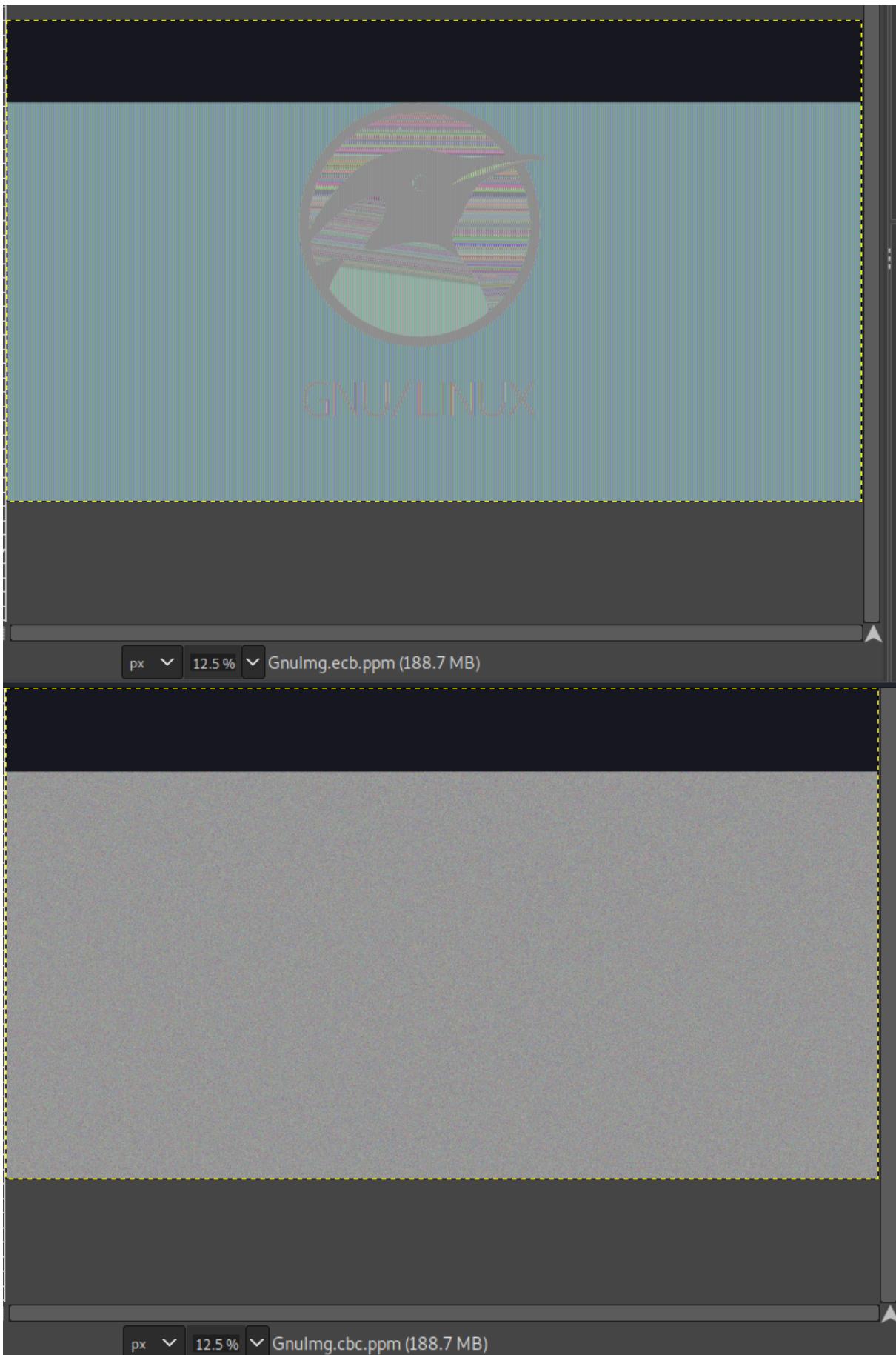
```
linux ~ head -n 4 kiwi.ppm>header.txt
head: cannot open 'kiwi.ppm' for reading: No such file or directory
linux ~ 1 cd Desktop/
linux ~ > Desktop head -n 4 kiwi.ppm>header.txt
linux ~ > Desktop tail -n +5 kiwi.ppm>body.bin
linux ~ > Desktop openssl enc -aes-128-ecb -nosalt -pass pass:"ANNA" -in body.bin -out body.ecb.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
linux ~ > Desktop cat header.txt body.ecb.bin>kiwi.ecb.ppm
linux ~ > Desktop gimp kiwi.ecb.ppm
linux ~ > Desktop openssl enc -aes-128-ecb -nosalt -pass pass:"ANNA" -in body.bin -out body.ecb.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
linux ~ > Desktop cat header.txt body.cbc.bin>kiwi.cbc.ppm
cat: body.cbc.bin: No such file or directory
linux ~ > Desktop 1 cat header.txt body.cbc.bin>kiwi.cbc.ppm
cat: body.cbc.bin: No such file or directory
linux ~ > Desktop 1 cat header.txt body.cbc.bin>kiwi.cbc.ppm
cat: body.cbc.bin: No such file or directory
linux ~ > Desktop 1 cat header2.txt body.cbc.bin>kiwi.cbc.ppm
cat: header2.txt: No such file or directory
cat: body.cbc.bin: No such file or directory
linux ~ > Desktop 1 cat header.txt body.cbc.bin>kiwi.cbc.ppm
cat: body.cbc.bin: No such file or directory
linux ~ > Desktop 1 head -n 4 kiwi.ppm>header2.txt
linux ~ > Desktop tail -n +5 kiwi.ppm>body2.bin
linux ~ > Desktop openssl enc -aes-128-ecb -nosalt -pass pass:"ANNA" -in body.bin -out body.ecb.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
linux ~ > Desktop 1 cat header.txt body.cbc.bin>kiwi.cbc.ppm
at: body.cbc.bin: No such file or directory
linux ~ > Desktop 1 head -n 4 kiwi.ppm>header2.txt
linux ~ > Desktop tail -n +5 kiwi.ppm>body2.bin
linux ~ > Desktop openssl enc -aes-128-ecb -nosalt -pass pass:"ANNA" -in body.bin -out body.ecb.bin
** WARNING : deprecated key derivation used.
sing -iter or -pbkdf2 would be better.
linux ~ > Desktop cat header.txt body.cbc.bin>kiwi.cbc.ppm
at: body.cbc.bin: No such file or directory
linux ~ > Desktop 1 cat header2.txt body.cbc.bin>kiwi.cbc.ppm
at: body.cbc.bin: No such file or directory
linux ~ > Desktop 1 cat header.txt body.cbc.bin>kiwi.cbc.ppm
at: header.txt: No such file or directory
at: body.cbc.bin: No such file or directory
linux ~ > Desktop 1
linux ~ > Desktop 1
linux ~ > Desktop 1
linux ~ > Desktop 1
linux ~ > Desktop 1
linux ~ > Desktop 1
linux ~ > Desktop 1 head -n 4 kiwi.ppm>header.txt
linux ~ > Desktop tail -n +5 kiwi.ppm>body.bin
linux ~ > Desktop openssl enc -aes-128-ecb -nosalt -pass pass:"ANNA" -in body.bin -out body.ecb.bin
** WARNING : deprecated key derivation used.
sing -iter or -pbkdf2 would be better.
linux ~ > Desktop cat header.txt body.ecb.bin>kiwi.ecb.ppm
linux ~ > Desktop gimp kiwi.ecb.ppm
linux ~ > Desktop openssl enc -aes-128-cbc -nosalt -pass pass:"ANNA" -in body.bin -out body.cbc.bin
** WARNING : deprecated key derivation used.
sing -iter or -pbkdf2 would be better.
linux ~ > Desktop cat header.txt body.cbc.bin>kiwi.cbc.ppm
linux ~ > Desktop
```



Bahadir Çetin Screenshots:

```
kali@kali:~$ openssl enc -aes-128-cbc -nosalt -pass pass:"ANNA" -in body.bin -out body.cbc.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
kali@kali:~$ cat header.txt body.cbc.bin > GnuImg.cbc.ppm
kali@kali:~$ gimp GnuImg.cbc.ppm
kali@kali:~$ openssl enc -aes-128-cbc -nosalt -pass pass:"ANNA" -in body.bin -out body.cbc.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
kali@kali:~$ cat header.txt body.cbc.bin > GnuImg.cbc.ppm
kali@kali:~$ gimp GnuImg.cbc.ppm
kali@kali:~$ ls
bah.txt      Downloads          Music      Videos
Cagatay.txt  Gnu-linux_minimalistic_logo.ppm Pictures
Desktop      Gnu-linux_minimalistic_logo.svg  Public
Documents    HW1               Templates
kali@kali:~$ head -n 4 Gnu-linux_minimalistic_logo.ppm > header.txt
kali@kali:~$ tail -n +5 Gnu-linux_minimalistic_logo.ppm > body.bin
kali@kali:~$ openssl enc -aes-128-ecb -nosalt -pass pass:"ANNA" -in body.bin -out body.ecb.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
kali@kali:~$ cat header.txt body.ecb.bin > GnuImg.ecb.ppm
kali@kali:~$ gimp GnuImg.ecb.ppm
```





Armağan OĞUZ Screenshots:

o SSL/TLS Client and Server Tests

Linux Lite Terminal -

Dosya Düzenle Görünüm Uçbirim Sekmeler Yardım

```
Welcome to Linux Lite 5.2 khrystal06sl, run the command
man openssl
Cuma 18 Aralık 2020, 07:49:07 ps://www.openssl.org/docs/manmaster/man1/openssl.html
Memory Usage: 1562/3936MB (39.68%)
Disk Usage: 7/20GB (42%) es available from http://www.madou.com/geo/openssl/
Support - https://www.linuxliteos.com/forums/ (Right click, Open Link)

To get the list of symmetric encryptions algorithms supported, run
khrystal06 ~ head -n 4 dragon.ppm > header.txt
head: cannot open 'dragon.ppm' for reading: No such file or directory
khrystal06 ~ 1 cd
khrystal06 ~ cd
khrystal06 ~ head -n 4 dragon.ppm > header.txt
head: cannot open 'dragon.ppm' for reading: No such file or directory
khrystal06 ~ 1 cd Desktop
khrystal06 ~ > Desktop head -n 4 dragon.ppm > header.txt
khrystal06 ~ > Desktop setxkbmap tr
khrystal06 ~ > Desktop tail -n +5 dragon.ppm > body.bin
khrystal06 ~ > Desktop openssl enc -aes-128-ecb -nosalt -pass pass:"ANNA" -
in body.bin -out body.ecb.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
khrystal06 ~ > Desktop cat header.txt body.ecb.bin > dragon.ecb.ppm
khrystal06 ~ > Desktop gimp dragon.ecb.ppm
GIMP-Warning: Error while parsing '/home/khrystal06/.config/GIMP/2.10/toolrc' in
Linux Lite Terminal - GIMP/2.10/toolrc' in
```

o SSL/TLS Client and Server Tests

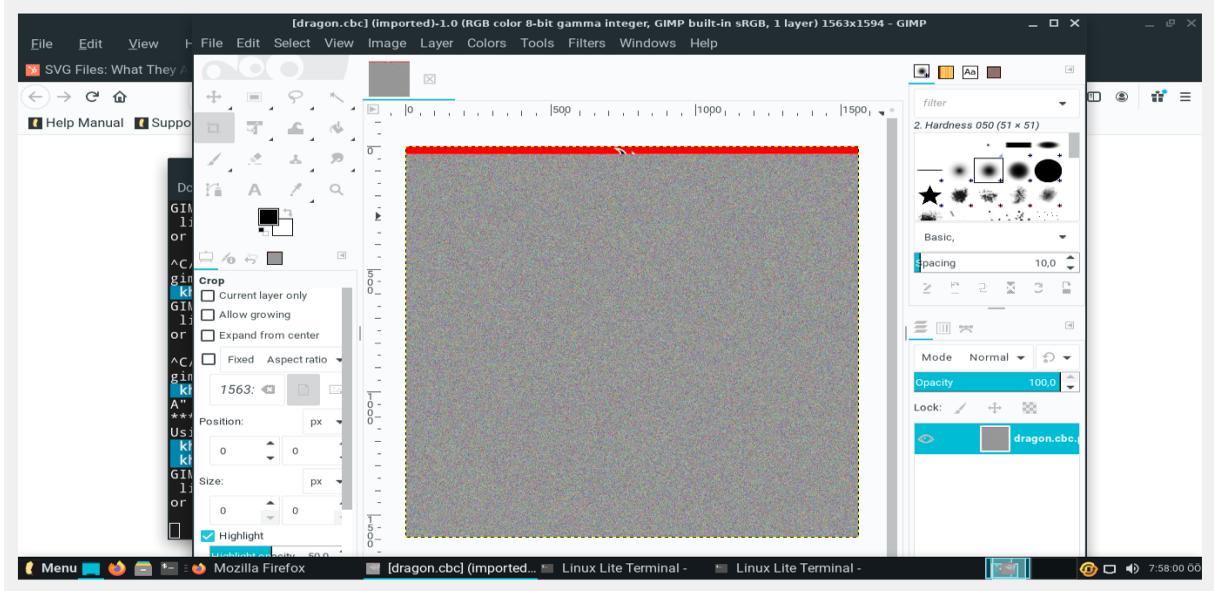
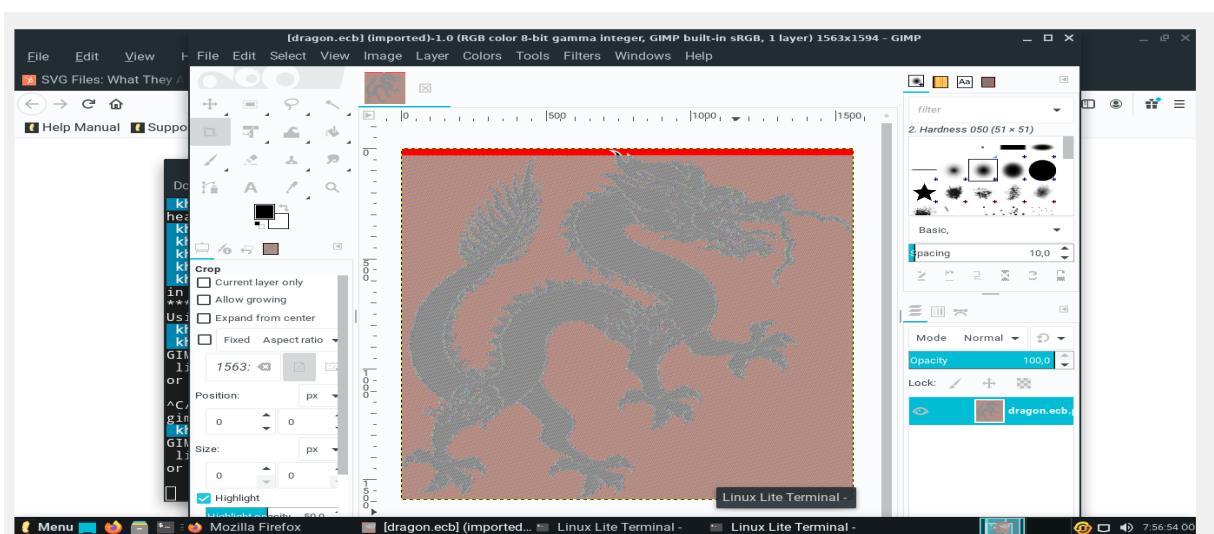
Linux Lite Terminal -

Dosya Düzenle Görünüm Uçbirim Sekmeler Yardım

```
or
For more information about Openssl, run the command
man openssl
^C/usr/lib/gimp/2.0/plug-ins/script-fu/script-fu terminated: Interrupt
gimp: terminated: Interrupt
khrystal06 ~ > Desktop 1 gimp dragon.ecb.ppm
GIMP-Warning: Error while parsing '/home/khrystal06/.config/GIMP/2.10/toolrc' in
line 3: unexpected identifier 'GimpToolInfo', expected symbol - fatal parse error
or

^C/usr/lib/gimp/2.0/plug-ins/script-fu/script-fu terminated: Interrupt
gimp: terminated: Interrupt
khrystal06 ~ > Desktop 1 openssl enc -aes-128-cbc d.Sect2 of the pass ipass:"ANN
A" -in body.bin -out body.cbc.bin pcamacho/tutorial/crypto/openssl/intro.html
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
File 105 MB file using the command
khrystal06 ~ > Desktop cat header.txt body.cbc.bin > dragon.cbc.ppm
khrystal06 ~ > Desktop gimp dragon.cbc.ppm
GIMP-Warning: Error while parsing '/home/khrystal06/.config/GIMP/2.10/toolrc' in
line 3: unexpected identifier 'GimpToolInfo', expected symbol - fatal parse error
or

^C/usr/lib/gimp/2.0/plug-ins/script-fu/script-fu terminated: Interrupt
gimp: terminated: Interrupt
khrystal06 ~ > Desktop 1
```



Modifying AES.java

Çağatay DOĞAN Screenshots:

CTR

The screenshot shows the IntelliJ IDEA interface with the project 'AES' open. The 'AES.java' file is the active editor. The code implements AES encryption using CTR mode. It reads a file named 'cagataydoran.txt', prints its content, encrypts it, and then decrypts it back to the original text. The run output window at the bottom shows the command run, the input file content, the encrypted binary output, and the decrypted output.

```
public class AES {
    static String IV = "UFUKCAGATAYDOGAN";
    static String plaintext = "test text 123\0\0\0"; /*Note null padding*/
    static String encryptionKey = "1996cagataydoran";

    public static String readFileAsString(String fileName) throws Exception {
        String data = "";
        data = new String(Files.readAllBytes(Paths.get(fileName)));
        return data;
    }

    public static void main(String [] args) throws IOException, Exception {
        String data = readFileAsString("cagataydoran.txt");
        System.out.println(data);
        try {

            System.out.println("==Java==");
            System.out.println("plain: " + data);

            byte[] cipher = encrypt(data, encryptionKey);

            System.out.print("cipher: ");
            for (int i=0; i<cipher.length; i++)
                System.out.print(new Integer(cipher[i])+" ");
            System.out.println(" ");

            String decrypted = decrypt(cipher, encryptionKey);

            System.out.println("decrypt: " + decrypted);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static byte[] encrypt(String plainText, String encryptionKey) throws Exception {
        Cipher cipher = Cipher.getInstance("AES/CTR/NoPadding", provider: "SunJCE");
        SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes( charsetName: "UTF-8"), algorithm: "AES");
        cipher.init(Cipher.ENCRYPT_MODE, key, new IvParameterSpec(IV.getBytes( charsetName: "UTF-8")));
        return cipher.doFinal(plainText.getBytes( charsetName: "UTF-8"));
    }

    public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception{
        Cipher cipher = Cipher.getInstance("AES/CTR/NoPadding", provider: "SunJCE");
        SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes( charsetName: "UTF-8"), algorithm: "AES");
        cipher.init(Cipher.DECRYPT_MODE, key, new IvParameterSpec(IV.getBytes( charsetName: "UTF-8")));
        return new String(cipher.doFinal(cipherText), charsetName: "UTF-8");
    }
}
```

Run: AES

```
/Users/cagataydoran/Library/Java/JavaVirtualMachines/openjdk-15.0.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=53096
Merhaba ben Cagatay Dorgan

==Java==
plain: Merhaba ben Cagatay Dorgan

cipher: -33 -105 -99 50 -28 -128 -7 -94 5 49 -84 101 68 32 -105 12 -58 123 10 85 108 -81 84 69 105 -28

decrypt: Merhaba ben Cagatay Dorgan

Process finished with exit code 0
```

OFB

The screenshot shows the IntelliJ IDEA interface with the project 'AES' open. The 'AES.java' file is the active editor, displaying Java code for an AES encryption program. The code reads a file 'cagataydoran.txt', encrypts its content using an IV and key, and then decrypts it back to the original text. The 'Run' tab at the bottom shows the command used to run the application and the resulting output, which includes the plain text 'Merhaba ben Cagatay Dogan', the ciphered byte array, and the decrypted text.

```
public class AES {
    static String IV = "UFUKCAGATAYDOGAN";
    static String plaintext = "test text 123\0\0\0"; /*Note null padding*/
    static String encryptionKey = "1996cagataydoran";

    public static String readFileAsString(String fileName) throws Exception {
        String data = "";
        data = new String(Files.readAllBytes(Paths.get(fileName)));
        return data;
    }

    public static void main(String [] args) throws IOException, Exception {
        String data = readFileAsString(fileName: "cagataydoran.txt");
        System.out.println(data);
        try {
            System.out.println("==Java==");
            System.out.println("plain: " + data);

            byte[] cipher = encrypt(data, encryptionKey);

            System.out.print("cipher: ");
            for (int i=0; i<cipher.length; i++)
                System.out.print(new Integer(cipher[i])+" ");
            System.out.println("");

            String decrypted = decrypt(cipher, encryptionKey);

            System.out.println("decrypt: " + decrypted);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static byte[] encrypt(String plainText, String encryptionKey) throws Exception {
        Cipher cipher = Cipher.getInstance(transformation: "AES/OFB/NoPadding", provider: "SunJCE");
        SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes(charsetName: "UTF-8"), algorithm: "AES");
        cipher.init(Cipher.ENCRYPT_MODE, key, new IvParameterSpec(IV.getBytes(charsetName: "UTF-8")));
        return cipher.doFinal(plainText.getBytes(charsetName: "UTF-8"));
    }

    public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception{
        Cipher cipher = Cipher.getInstance(transformation: "AES/OFB/NoPadding", provider: "SunJCE");
        SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes(charsetName: "UTF-8"), algorithm: "AES");
        cipher.init(Cipher.DECRYPT_MODE, key, new IvParameterSpec(IV.getBytes(charsetName: "UTF-8")));
        return new String(cipher.doFinal(cipherText), charsetName: "UTF-8");
    }
}
```

Run: AES

```
/Users/cagataydoran/Library/Java/JavaVirtualMachines/openjdk-15.0.1/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=53106
Merhaba ben Cagatay Dogan

==Java==
plain: Merhaba ben Cagatay Dogan

cipher: -33 -105 -99 50 -28 -120 -7 -94 5 49 -84 101 68 32 -105 12 -54 99 -54 -75 -37 106 96 69 41 96

decrypt: Merhaba ben Cagatay Dogan

Process finished with exit code 0
```

ECB

The screenshot shows the IntelliJ IDEA interface with a Java project named "AES". The code editor displays "AES.java" which contains an AES implementation using ECB mode. The console output shows the encryption and decryption of the string "Merhaba ben Cagatay Dogan".

```
import javax.crypto.spec.SecretKeySpec;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

public class AES {
    static String IV = "UEUKCAGATAYDOGAN";
    static String plaintext = "test text 123\0\0\0"; /*Note null padding*/
    static String encryptionKey = "1996cagataydogen";

    public static String readFileAsString(String fileName) throws Exception {
        String data = "";
        data = new String(Files.readAllBytes(Paths.get(fileName)));
        return data;
    }

    public static void main(String [] args) throws IOException, Exception {
        String data = readFileAsString( fileName: "cagataydogen.txt");
        System.out.println(data);
        try {

            System.out.println("==Java==");
            System.out.println("plain: " + data);

            byte[] cipher = encrypt(data, encryptionKey);
        }
    }
}
```

Run: AES

```
plain: Merhaba ben Cagatay Dogan
cipher: -125 80 -109 -28 -76 -91 35 -17 76 -96 -90 33 -1 -93 -122 -105 -40 11 -45 -98 -6 2 -109 -111 56 -105 -26 -50 -27 20 8 56
decrypt: Merhaba ben Cagatay Dogan

Process finished with exit code 0
```

Event Log: 63:50 LF UTF-8 2 spaces* 8ms

The screenshot shows the IntelliJ IDEA interface with a Java project named "AES". The code editor displays "AES.java" which contains an AES implementation using ECB mode with detailed comments. The console output shows the encryption and decryption of the string "Merhaba ben Cagatay Dogan".

```
System.out.print("cipher: ");
for (int i=0; i<cipher.length; i++)
    System.out.print(new Integer(cipher[i])+" ");
System.out.println("");

String decrypted = decrypt(cipher, encryptionKey);

System.out.println("decrypt: " + decrypted);

} catch (Exception e) {
    e.printStackTrace();
}

public static byte[] encrypt(String plainText, String encryptionKey) throws Exception {
    Cipher cipher = Cipher.getInstance( transformation: "AES/ECB/PKCS5Padding", provider: "SunJCE");
    SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes( charsetName: "UTF-8"), algorithm: "AES");
    cipher.init(Cipher.ENCRYPT_MODE, key);
    return cipher.doFinal(plainText.getBytes( charsetName: "UTF-8"));
}

public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception{
    Cipher cipher = Cipher.getInstance( transformation: "AES/ECB/PKCS5Padding", provider: "SunJCE");
    SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes( charsetName: "UTF-8"), algorithm: "AES");
    return new String(cipher.doFinal(cipherText), charsetName: "UTF-8");
}
```

Run: AES

```
plain: Merhaba ben Cagatay Dogan
cipher: -125 80 -109 -28 -76 -91 35 -17 76 -96 -90 33 -1 -93 -122 -105 -40 11 -45 -98 -6 2 -109 -111 56 -105 -26 -50 -27 20 8 56
decrypt: Merhaba ben Cagatay Dogan

Process finished with exit code 0
```

Event Log: 63:50 LF UTF-8 2 spaces* 8ms

Bahadir ÇETİN Screenshots:

CTR

The screenshot shows the NetBeans IDE 8.2 interface with the following details:

- Project Explorer:** Shows the project "AES" with the file "AES.java" selected.
- Code Editor:** Displays the Java code for AES encryption using CTR mode. The code includes imports for javax.crypto.Cipher, javax.crypto.spec.IvParameterSpec, and javax.crypto.spec.SecretKeySpec. It defines a static IV, a static plaintext string, and an encryption key. The main method reads a file, prints the plain text, encrypts it, prints the cipher bytes, decrypts it, and prints the decrypted text.
- Output Window:** Shows the terminal output of the run command. It includes the plain text "Hello its bahadir how are you?", the generated cipher bytes (-86 113 45 42 -77 28 112 -52 11 1 -108 -9 -99 -87 101 61 -28 -24 4 81 -30 72 2 60 -48 95 10 105 -57 -55), the decrypted text "Hello its bahadir how are you?", and the message "BUILD SUCCESSFUL (total time: 1 second)".

```
13 import javax.crypto.Cipher;
14 import javax.crypto.spec.IvParameterSpec;
15 import javax.crypto.spec.SecretKeySpec;
16
17
18 public class AES {
19     static String IV = "AAAAAAAAAAAAAA";
20     static String plaintext = "test text 123\0\0\0"; //Note null padding
21     static String encryptionKey = "0123456789asdqwe";
22
23     public static String readFileAsString(String fileName) throws Exception {
24         String data = "";
25         data = new String(Files.readAllBytes(Paths.get(fileName)));
26         return data;
27     }
28
29
30     public static void main(String [] args) throws IOException, Exception {
31         String data = readFileAsString("C:\\\\Users\\\\rog51\\\\Desktop\\\\BahadirCetinProject.txt");
32         System.out.println(data);
33         try {
34
35             System.out.println("==Java==");
36             System.out.println("plain: " + data);
37
38             byte[] cipher = encrypt(data, encryptionKey);
39
40             System.out.print("cipher: ");
41             for (int i=0; i<cipher.length; i++)
42                 System.out.print(new Integer(cipher[i])+" ");
43             System.out.println("");
44
45             String decrypted = decrypt(cipher, encryptionKey);
46
47             System.out.println("decrypt: " + decrypted);
48
49         } catch (Exception e) {
50             e.printStackTrace();
51         }
52     }
53
54     public static byte[] encrypt(String plainText, String encryptionKey) throws Exception {
55         Cipher cipher = Cipher.getInstance("AES/CTR/NoPadding", "SunJCE");
56         SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
57         cipher.init(Cipher.ENCRYPT_MODE, key,new IvParameterSpec(IV.getBytes("UTF-8")));
58         return cipher.doFinal(plainText.getBytes("UTF-8"));
59     }
60
61     public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception{
62         Cipher cipher = Cipher.getInstance("AES/CTR/NoPadding", "SunJCE");
63         SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
64         cipher.init(Cipher.DECRYPT_MODE, key,new IvParameterSpec(IV.getBytes("UTF-8")));
65         return new String(cipher.doFinal(cipherText),"UTF-8");
66     }
67 }
```

Output - AES (run)

```
run:
Hello its bahadir how are you?
==Java==
plain: Hello its bahadir how are you?
cipher: -86 113 45 42 -77 28 112 -52 11 1 -108 -9 -99 -87 101 61 -28 -24 4 81 -30 72 2 60 -48 95 10 105 -57 -55
decrypt: Hello its bahadir how are you?
BUILD SUCCESSFUL (total time: 1 second)
```

OFB

The screenshot shows the NetBeans IDE 8.2 interface with the following details:

- Title Bar:** AES - NetBeans IDE 8.2
- Menu Bar:** File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
- Toolbar:** Standard NetBeans toolbar icons.
- Project Explorer:** Shows the project structure under the AES node:
 - Source Packages: aes (containing AES.java)
 - Libraries
 - FiroMidterm
 - FiroQuiz
 - HotelBookingApplication
 - Seyit_Bahadir_Cetin
 - UMLDiagrams
- Code Editor:** Displays the AES.java source code. The code implements AES encryption and decryption using the Cipher class in Java's crypto API. It includes imports for javax.crypto.Cipher, javax.crypto.spec.IvParameterSpec, and javax.crypto.spec.SecretKeySpec. The main method reads a file named "BahirCetinProject.txt", prints its content, encrypts it using an IV of "AAAAAAAAAAAAAAA", and then decrypts it back to its original form.
- Output Window:** Shows the terminal output of the run command:

```
run:
Hello its bahadir how are you?
==Java==
plain: Hello its bahadir how are you?
cipher: -86 113 45 42 -77 28 112 -52 11 1 -108 -9 -99 -87 101 61 -115 -4 19 -44 89 -3 -15 23 31 -80 68 76 -81 55
decrypt: Hello its bahadir how are you?
BUILD SUCCESSFUL (total time: 1 second)
```
- Bottom Navigation:** Bookmarks, Output, Notifications.

ECB

```
import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

public class AES {
    static String IV = "AAAAAAAAAAAAAA";
    static String plaintext = "test text 123\0\0\0"; /*Note null padding*/
    static String encryptionKey = "0123456789asdqwe";

    public static String readFileAsString(String fileName) throws Exception
    {
        String data = "";
        data = new String(Files.readAllBytes(Paths.get(fileName)));
        return data;
    }

    public static void main(String [] args) throws IOException, Exception {
        String data = readFileAsString("C:\\Users\\rog91\\Desktop\\BahadirCetinProject.txt");
        System.out.println(data);
        try {

            System.out.println("==Java==");
            System.out.println("plain: " + data);

            byte[] cipher = encrypt(data, encryptionKey);

            System.out.print("cipher: ");
            for (int i=0; i<cipher.length; i++)
                System.out.print(new Integer(cipher[i])+" ");
            System.out.println("");

            String decrypted = decrypt(cipher, encryptionKey);

            System.out.println("decrypt: " + decrypted);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static byte[] encrypt(String plainText, String encryptionKey) throws Exception {
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding", "SunJCE");
        SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
        cipher.init(Cipher.ENCRYPT_MODE, key);
        return cipher.doFinal(plainText.getBytes("UTF-8"));
    }

    public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception{
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding", "SunJCE");
        SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
        cipher.init(Cipher.DECRYPT_MODE, key);
        return new String(cipher.doFinal(cipherText),"UTF-8");
    }
}
```

```
Output - AES (run)
▶ run:
▶ Hello its bahadir how are you?
==Java==
plain: Hello its bahadir how are you?
cipher: -27 120 -53 -27 61 -69 -41 -49 -77 125 -90 -90 -61 67 -121 -28 39 55 72 -40 -11 83 1 80 -70 -47 102 108 -21 65 3 24
decrypt: Hello its bahadir how are you?
BUILD SUCCESSFUL (total time: 1 second)
```

Berk ÖNDER Screenshots:

CTR

```
51  public static byte[] encrypt(String plainText, String encryptionKey) throws Exception {  
52      Cipher cipher = Cipher.getInstance("AES/CTR/NoPadding", "SunJCE");  
53      SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");  
54      cipher.init(Cipher.ENCRYPT_MODE, key,new IvParameterSpec(IV.getBytes("UTF-8")));  
55      return cipher.doFinal(plainText.getBytes("UTF-8"));  
56  }  
57  
58  public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception{  
59      Cipher cipher = Cipher.getInstance("AES/CTR/NoPadding", "SunJCE");  
60      SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");  
61      cipher.init(Cipher.DECRYPT_MODE, key,new IvParameterSpec(IV.getBytes("UTF-8")));  
62      return new String(cipher.doFinal(cipherText),"UTF-8");  
63  }  
64 }
```

OFB

```
51  public static byte[] encrypt(String plainText, String encryptionKey) throws Exception {  
52      Cipher cipher = Cipher.getInstance("AES/OFB/NoPadding", "SunJCE");  
53      SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");  
54      cipher.init(Cipher.ENCRYPT_MODE, key,new IvParameterSpec(IV.getBytes("UTF-8")));  
55      return cipher.doFinal(plainText.getBytes("UTF-8"));  
56  }  
57  
58  public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception{  
59      Cipher cipher = Cipher.getInstance("AES/OFB/NoPadding", "SunJCE");  
60      SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");  
61      cipher.init(Cipher.DECRYPT_MODE, key,new IvParameterSpec(IV.getBytes("UTF-8")));  
62      return new String(cipher.doFinal(cipherText),"UTF-8");  
63  }  
64 }
```

OUTPUT

Output - Running Single Java File X

```
Note: C:\Users\Berk\Desktop\OpenSSL-tutorial\AES.java uses or overrides a deprecated API.  
Note: Recompile with -Xlint:deprecation for details.  
Berk Onder ben  
==Java==  
plain: Berk Onder ben  
cipher: 121 -104 118 -89 45 49 -69 55 61 -112 -21 -125 -8 -116  
decrypt: Berk Onder ben
```

CODE

```
15  public class AES {
16      static String IV = "AAAAAAAAAAAAAAA";
17      static String plaintext = "test text 123\u0000"; /*Note null padding*/
18      static String encryptionKey = "0123456789abcdef";
19
20      public static String readFileAsString(String fileName) throws Exception
21      {
22          String data = "";
23          data = new String(Files.readAllBytes(Paths.get(fileName)));
24          return data;
25      }
26
27      public static void main(String [] args) throws IOException, Exception {
28          String data = readFileAsString("C:/Users/Berk/Desktop/OpenSSL-tutorial/proje.txt");
29          System.out.println(data);
30          try {
31
32              System.out.println("==Java==");
33              System.out.println("plain: " + data);
34
35              byte[] cipher = encrypt(data, encryptionKey);
36
37              System.out.print("cipher: ");
38              for (int i=0; i<cipher.length; i++)
39                  System.out.print(new Integer(cipher[i])+" ");
40              System.out.println("");
41
42              String decrypted = decrypt(cipher, encryptionKey);
43
44              System.out.println("decrypt: " + decrypted);
45
46          } catch (Exception e) {
47              e.printStackTrace();
48          }
49      }
}
```

ECB

```
15 public class AES {
16     static String IV = "AAAAAAAAAAAAAA";
17     static String plaintext = "test text 123\0\0\0"; /*Note null padding*/
18     static String encryptionKey = "0123456789abcdef";
19
20     public static String readFileAsString(String fileName) throws Exception
21     {
22         String data = "";
23         data = new String(Files.readAllBytes(Paths.get(fileName)));
24         return data;
25     }
26
27     public static void main(String [] args) throws IOException, Exception {
28         String data = readFileAsString("C:/Users/Berk/Desktop/OpenSSL-tutorial/proje.txt");
29         System.out.println(data);
30         try {
31
32             System.out.println("==Java==");
33             System.out.println("plain: " + data);
34
35             byte[] cipher = encrypt(data, encryptionKey);
36
37             System.out.print("cipher: ");
38             for (int i=0; i<cipher.length; i++)
39                 System.out.print(new Integer(cipher[i])+" ");
40             System.out.println("");
41
42             String decrypted = decrypt(cipher, encryptionKey);
43
44             System.out.println("decrypt: " + decrypted);
45
46         } catch (Exception e) {
47             e.printStackTrace();
48         }
49     }
50
51     public static byte[] encrypt(String plainText, String encryptionKey) throws Exception {
52         Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding", "SunJCE");
53         SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
54         cipher.init(Cipher.ENCRYPT_MODE, key);
55         return cipher.doFinal(plainText.getBytes("UTF-8"));
56     }
57
58     public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception{
59         Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding", "SunJCE");
60         SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
61         cipher.init(Cipher.DECRYPT_MODE, key);
62         return new String(cipher.doFinal(cipherText),"UTF-8");
63     }
64 }
```

Output - Running Single Java File X

```
Note: C:\Users\Berk\Desktop\OpenSSL-tutorial\AES.java uses or overrides a deprecated API.  
Note: Recompile with -Xlint:deprecation for details.  
Berk Onder  
==Java==  
plain: Berk Onder  
cipher: -122 51 -63 -96 5 66 104 -120 -52 55 120 0 -19 36 -82 -14  
decrypt: Berk Onder
```

Armağan OĞUZ Screenshots:

CTR

```
public class AES {
    static String IV = "AAAAAAAAAAAAAAA";
    static String plaintext = "test text 123\0\0\0"; /*Note null padding*/
    static String encryptionKey = "0123456789abcdef";

    public static String readFileAsString(String fileName) throws Exception
    {
        String data = "";
        data = new String(Files.readAllBytes(Paths.get(fileName)));
        return data;
    }

    public static void main(String [] args) throws IOException, Exception {
        String data = readFileAsString("C:/Users/khrystal06/Desktop/dragon.txt");
        System.out.println(data);
        try {

            System.out.println("==Java==");
            System.out.println("plain: " + data);

            byte[] cipher = encrypt(data, encryptionKey);

            System.out.print("cipher: ");
            for (int i=0; i<cipher.length; i++)
                System.out.print(new Integer(cipher[i])+" ");
            System.out.println("");

            String decrypted = decrypt(cipher, encryptionKey);

            System.out.println("decrypt: " + decrypted);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static byte[] encrypt(String plainText, String encryptionKey) throws Exception {
        Cipher cipher = Cipher.getInstance("AES/CTR/NoPadding", "SunJCE");
        SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
        cipher.init(Cipher.ENCRYPT_MODE, key,new IvParameterSpec(IV.getBytes("UTF-8")));
        return cipher.doFinal(plainText.getBytes("UTF-8"));
    }

    public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception{
        Cipher cipher = Cipher.getInstance("AES/CTR/NoPadding", "SunJCE");
        SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
        cipher.init(Cipher.DECRYPT_MODE, key,new IvParameterSpec(IV.getBytes("UTF-8")));
    }
}
```

OFB

```
public class AES {
    static String IV = "AAAAAAAAAAAAAA";
    static String plaintext = "test text 123\0\0\0"; /*Note null padding*/
    static String encryptionKey = "0123456789abcdef";

    public static String readFileAsString(String fileName) throws Exception
    {
        String data = "";
        data = new String(Files.readAllBytes(Paths.get(fileName)));
        return data;
    }

    public static void main(String [] args) throws IOException, Exception {
        String data = readFileAsString("C:/Users/khrystral06/Desktop/dragon.txt");
        System.out.println(data);
        try {

            System.out.println("==Java==");
            System.out.println("plain: " + data);

            byte[] cipher = encrypt(data, encryptionKey);

            System.out.print("cipher: ");
            for (int i=0; i<cipher.length; i++)
                System.out.print(new Integer(cipher[i])+" ");
            System.out.println("");

            String decrypted = decrypt(cipher, encryptionKey);

            System.out.println("decrypt: " + decrypted);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static byte[] encrypt(String plainText, String encryptionKey) throws Exception {
        Cipher cipher = Cipher.getInstance("AES/OFB/NoPadding", "SunJCE");
        SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
        cipher.init(Cipher.ENCRYPT_MODE, key,new IvParameterSpec(IV.getBytes("UTF-8")));
        return cipher.doFinal(plainText.getBytes("UTF-8"));
    }

    public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception{
        Cipher cipher = Cipher.getInstance("AES/OFB/NoPadding", "SunJCE");
        SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
        cipher.init(Cipher.DECRYPT_MODE, key,new IvParameterSpec(IV.getBytes("UTF-8")));
    }
}
```

OUTPUT

```
Sevki Armagan
==Java==
plain: Sevki Armagan
cipher: 104 -104 114 -89 100 94 -108 33 53 -125 -84 -128 -13
decrypt: Sevki Armagan
```

ECB

The screenshot shows an IDE interface with multiple tabs open. The main tab contains Java code for an ECB mode cipher. The code includes methods for reading files, printing plain text, encrypting, and decrypting. It uses the Cipher class with AES/ECB/PKCS5Padding and SecretKeySpec.

```
16 import javax.crypto.spec.SecretKeySpec;
17
18 public class AES {
19     static String IV = "AAAAAAAAAAAAAA";
20     static String plaintext = "test text 123\0\0\0"; /*Note null padding*/
21     static String encryptionKey = "0123456789qweerty";
22
23     public static String readFileAsString(String fileName) throws Exception {
24         String data = "";
25         data = new String(Files.readAllBytes(Paths.get(fileName)));
26         return data;
27     }
28
29
30     public static void main(String [] args) throws IOException, Exception {
31         String data = readFileAsString(":\\Users\\khrystral\\Desktop\\AES\\src\\aes\\aramaganoguz.txt");
32         System.out.println(data);
33         try {
34
35             System.out.println("==Java==");
36             System.out.println("plain: " + data);
37
38             byte[] cipher = encrypt(data, encryptionKey);
39
40             System.out.print("cipher: ");
41             for (int i=0; i<cipher.length; i++)
42                 System.out.print(new Integer(cipher[i])+" ");
43             System.out.println("");
44
45             String decrypted = decrypt(cipher, encryptionKey);
46
47             System.out.println("decrypt: " + decrypted);
48
49         } catch (Exception e) {
50             e.printStackTrace();
51         }
52     }
53
54     public static byte[] encrypt(String plainText, String encryptionKey) throws Exception {
55         Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding", "SunJCE");
56         SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
57         cipher.init(Cipher.ENCRYPT_MODE, key);
58         return cipher.doFinal(plainText.getBytes("UTF-8"));
59     }
60
61     public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception {
62         Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding", "SunJCE");
63         SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
64         cipher.init(Cipher.DECRYPT_MODE, key);
65         return new String(cipher.doFinal(cipherText), "UTF-8");
66     }
67 }
```

The screenshot shows the Output window of the IDE. It displays the command-line output of the Java application. The application prints a message, reads a file, prints its content, performs encryption, and then decryption, comparing the original and decrypted strings.

```
I'm seeking armanoguz and I'm about to graduate.
==Java==
plain: hev, I'm seeking armanoguz and I'm about to graduate.
cipher: -75 29 -41 -156 89 124 -115 97 -79 126 -113 -69 106 -67 -54 -64 121 -26 -100 10 -114 -70 74 -27 -45 -21 117 -68 52 45 -37 -42 19 48 14 -100 -17 -60 122 81 -128 -91 85 -109 -19 -67 -94 -32 23 117 7 33 60 -88 50 -70 58 90 125 95 102 -128 -59
BUILD SUCCESSFUL (total time: 0 seconds)
```

5. Part of The Project: Secure Code Standards

Our secure coding implementation which is written by using 4 different rules by each group member can be found inside the ZIP file that we sent to you. We preferred Java as a programming language.

References:

<https://wiki.sei.cmu.edu/confluence/display/java/SEI+CERT+Oracle+Coding+Standard+for+Java>

<https://www.digitalocean.com/community/tutorials/how-to-create-a-ssl-certificate-on-apache-for-ubuntu-14-04>

<https://www.digitalocean.com/community/tutorials/how-to-create-a-ssl-certificate-on-apache-for-debian-8>

http://kb.mozilla.org/Installing_an_SMIME_certificate

<https://www.actalis.it/en/certificates-for-secure-electronic-mail.aspx>

<https://www.ssl.com/how-to/installing-an-s-mime-certificate-and-sending-secure-email-in-mozilla-thunderbird-on-windows-10/>

<https://www.linuxliteos.com/download.php>

SVG Images:

Berk Önder: <https://css-tricks.com/understanding-and-manually-improving-svg-optimization>

Çağatay Doğan: <https://commons.wikimedia.org/wiki/File:TUX-G2-SVG.svg>

Bahadir Çetin: https://commons.wikimedia.org/wiki/File:Gnu-linux_minimalistic_logo.svg

Armağan Oğuz:

https://commons.wikimedia.org/wiki/File:Chinese_black_dragon_red_background.svg