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

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

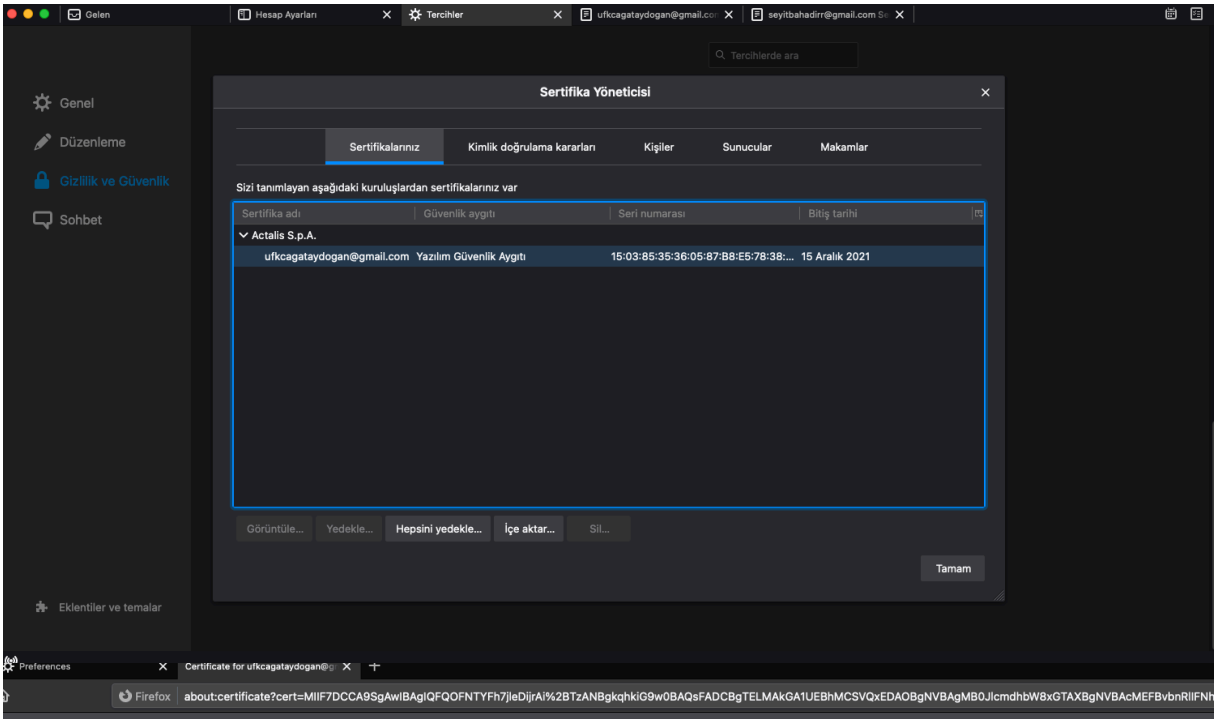
Key Size 2048

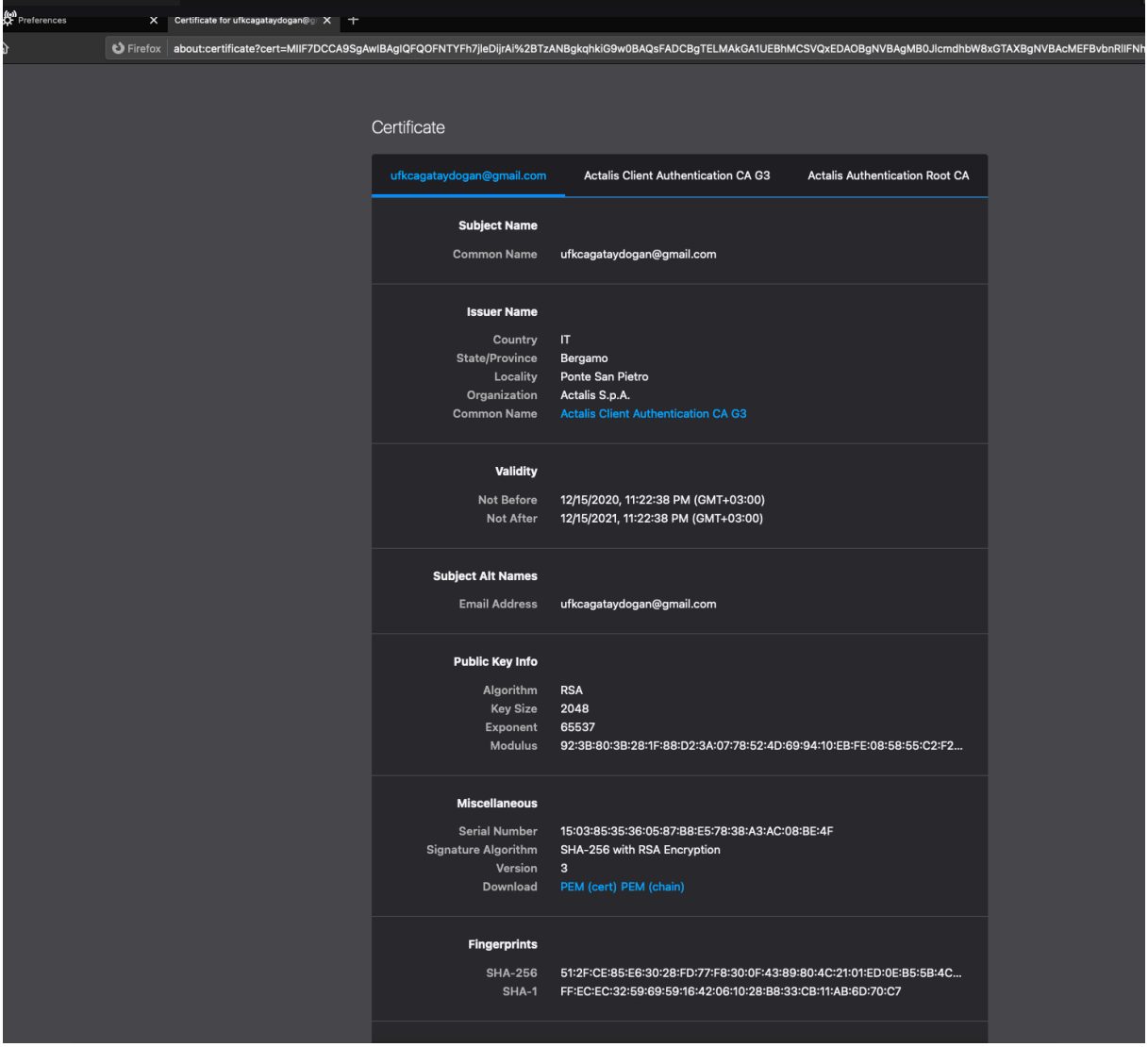
Exponent 65537

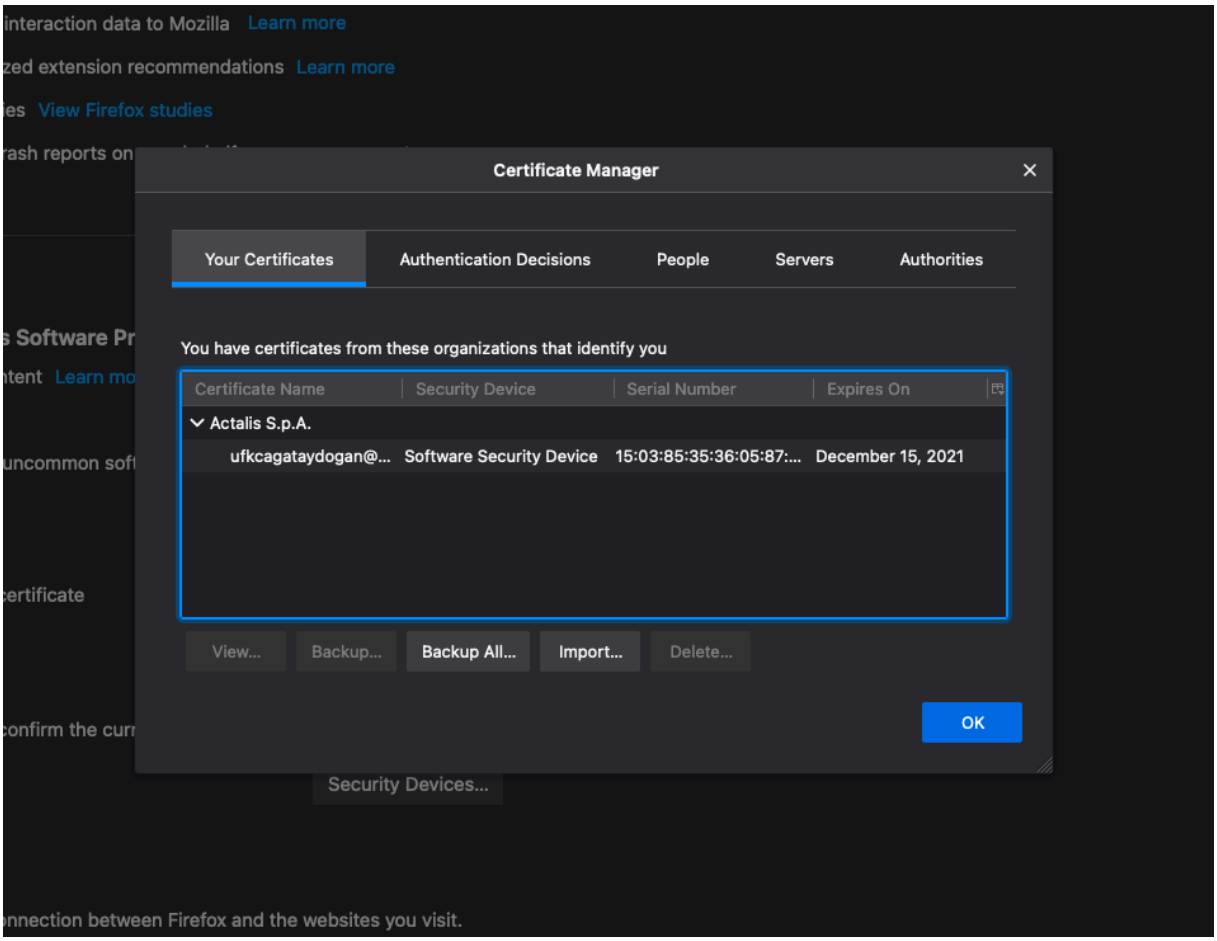
Modulus

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:






interaction data to Mozilla Learn more
zed extension recommendations Learn more
es View Firefox studies
rash reports on
Software Pr
ntent Learn mo
uncommon soft
ertificate
confirm the curr
nnection between Firefox and the websites you visit.
Certificate Manager
Your Certificates
Authentication Decisions
People
Servers
Authorities
You have certificates from these organizations that identify you
Certificate Name
Security Device
Serial Number
Expires On
Actalis S.p.A.
ufkcagataydogan@...
Software Security Device
15:03:85:35:36:05:87:...
December 15, 2021
View...
Backup...
Backup All...
Import...
Delete...
OK
Security Devices...

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

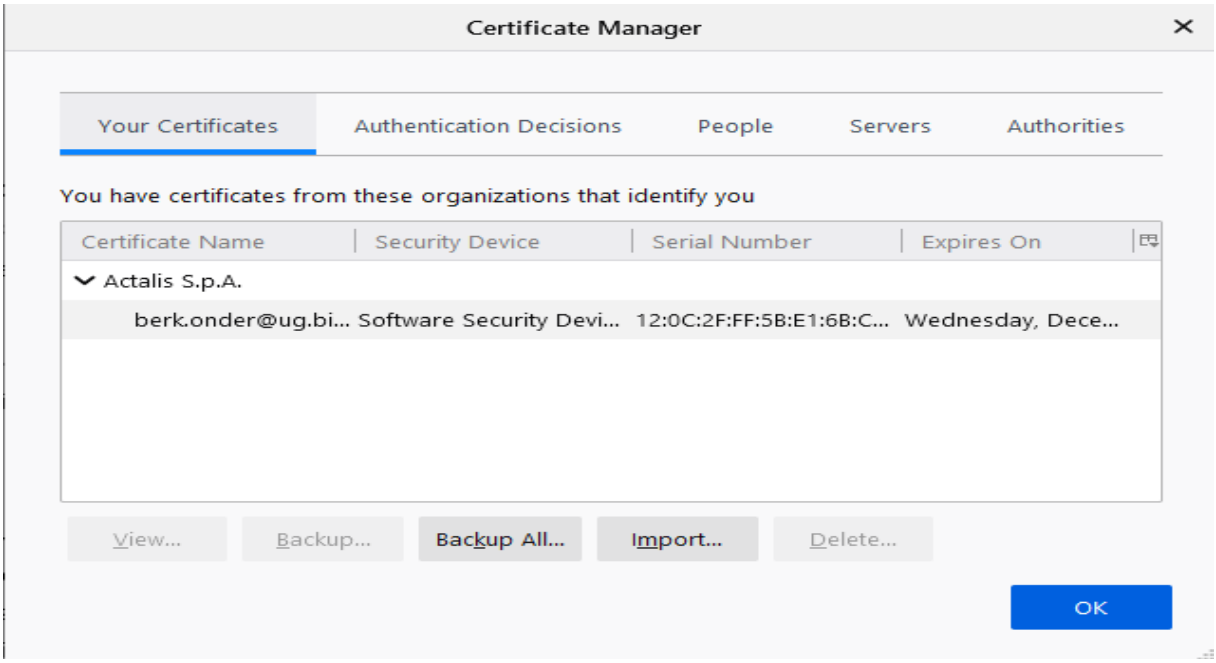
Key Size 2048

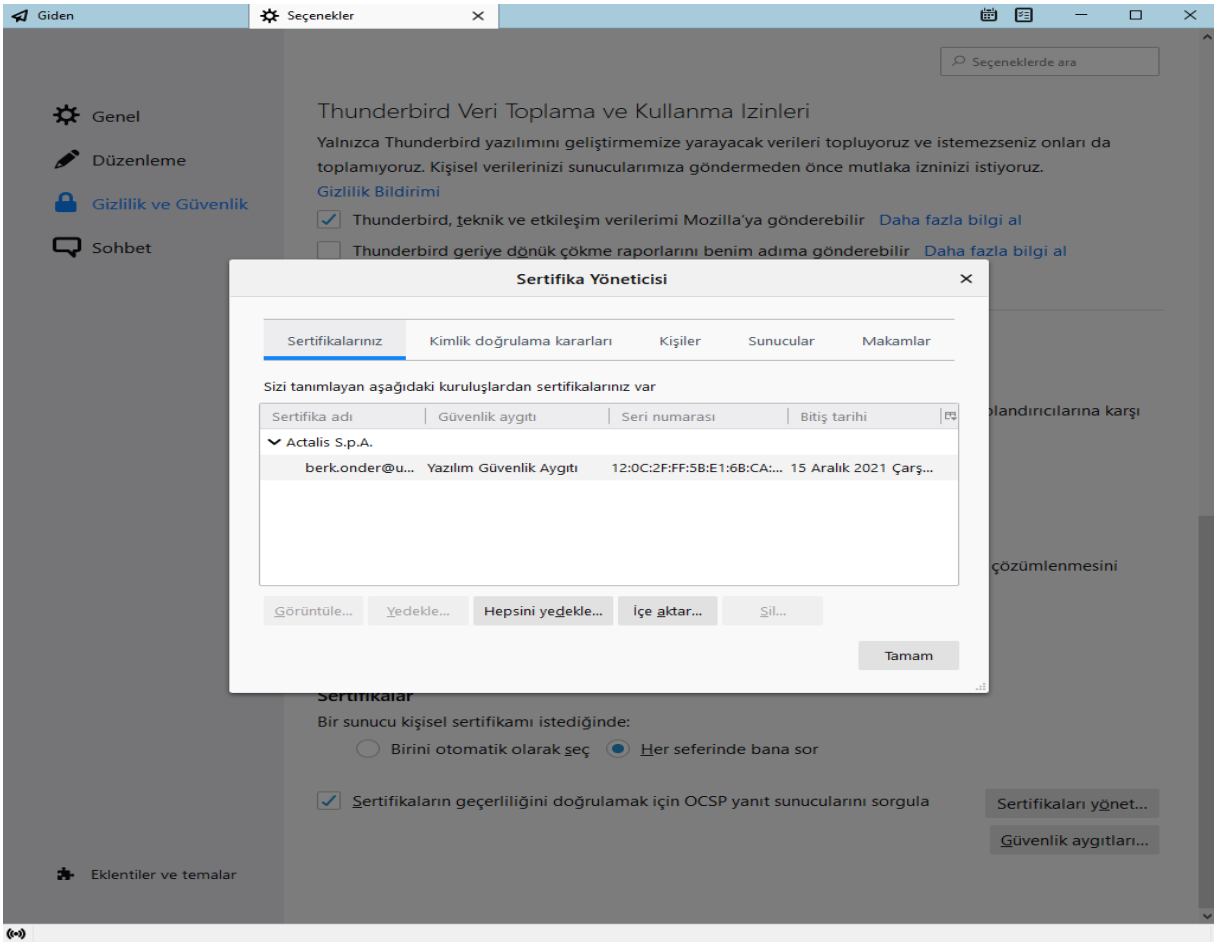
Exponent 65537

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

Name Surname: Seyit Bahadır 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

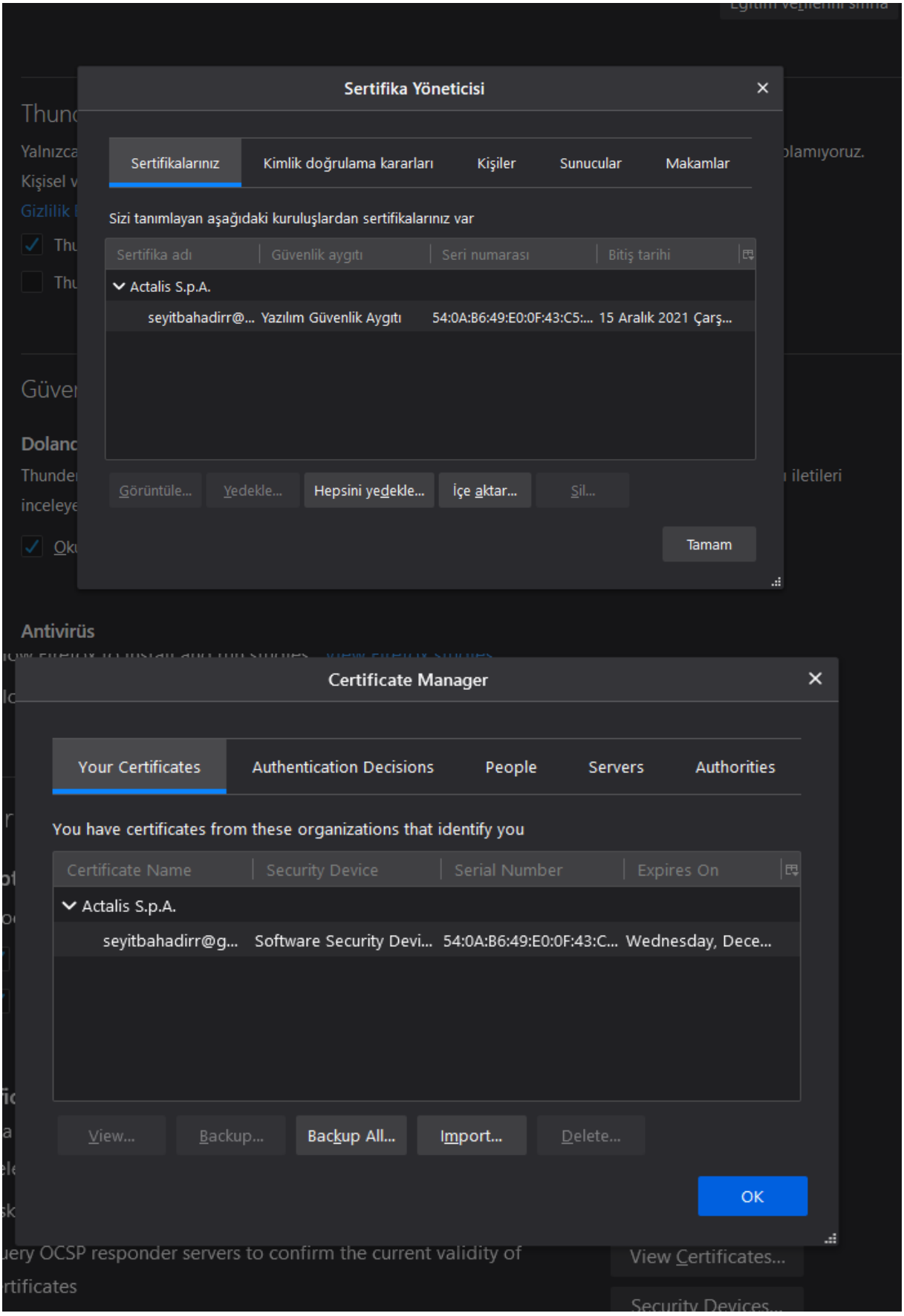
Key Size 2048

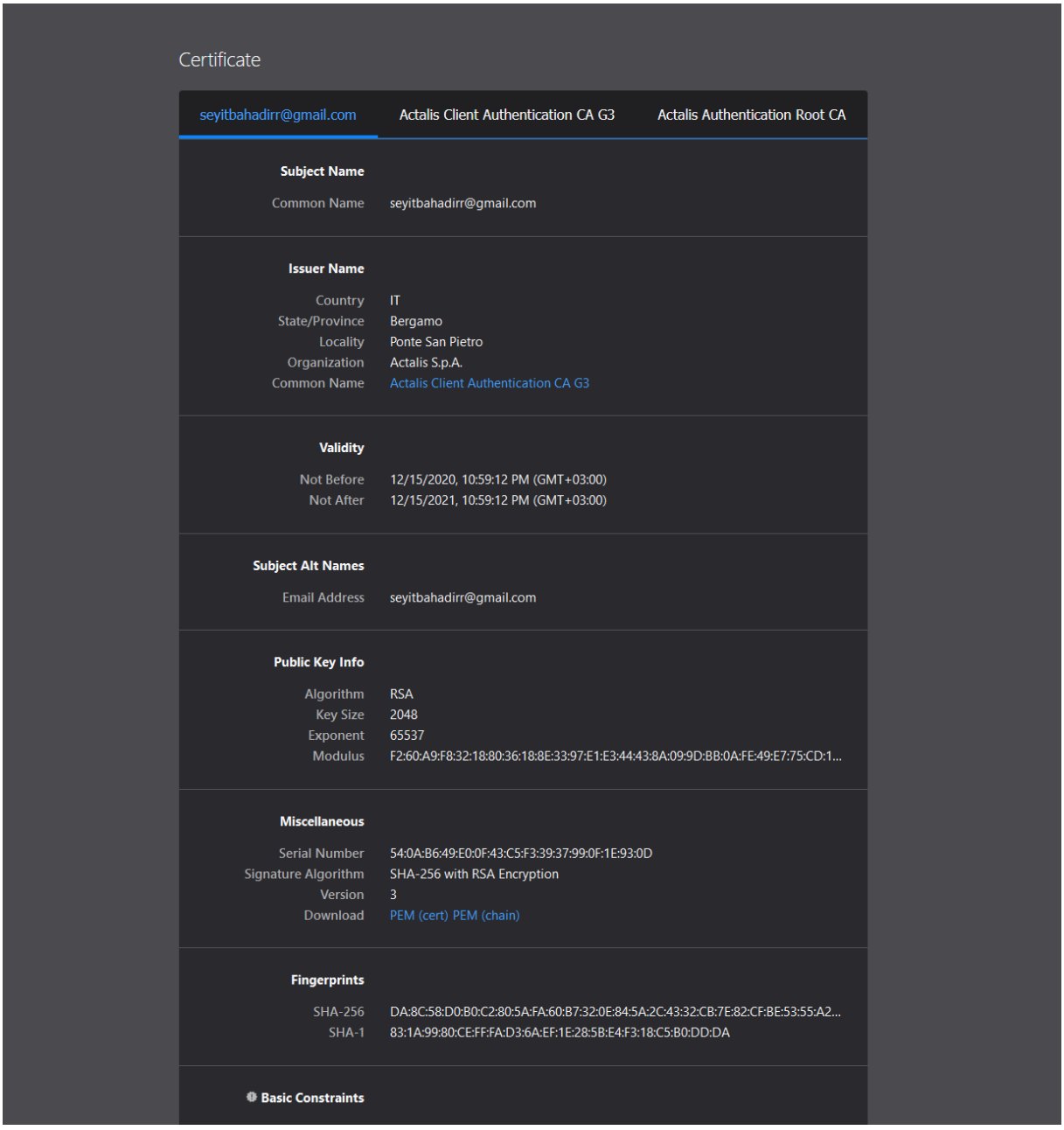
Exponent 65537

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

Bahadır ÇETİN Screenshots:




Certificate
seyitbahadirr@gmail.com
Actalis Client Authentication CA G3
Actalis Authentication Root CA
Subject Name
Common Name seyitbahadirr@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 seyitbahadirr@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:BB:0A:FE:49:E7:75:CD:1...
Miscellaneous
Serial Number 54:0A:B6:49:E0:0F:43: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

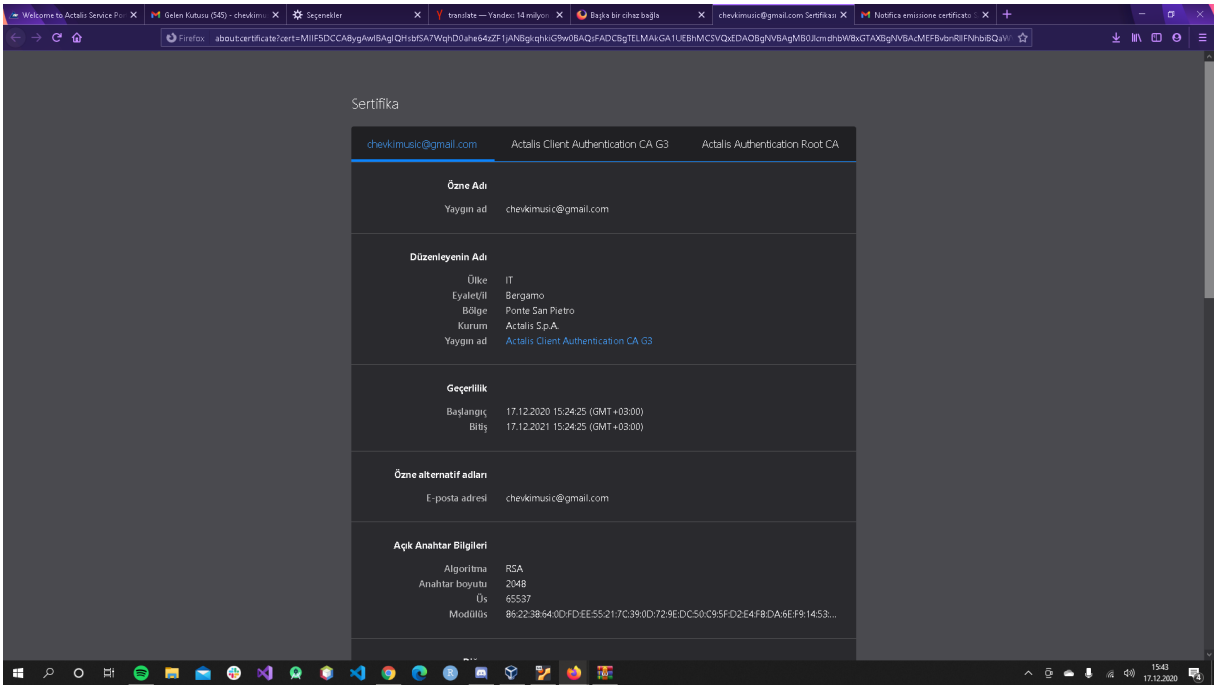
Key Size 2048

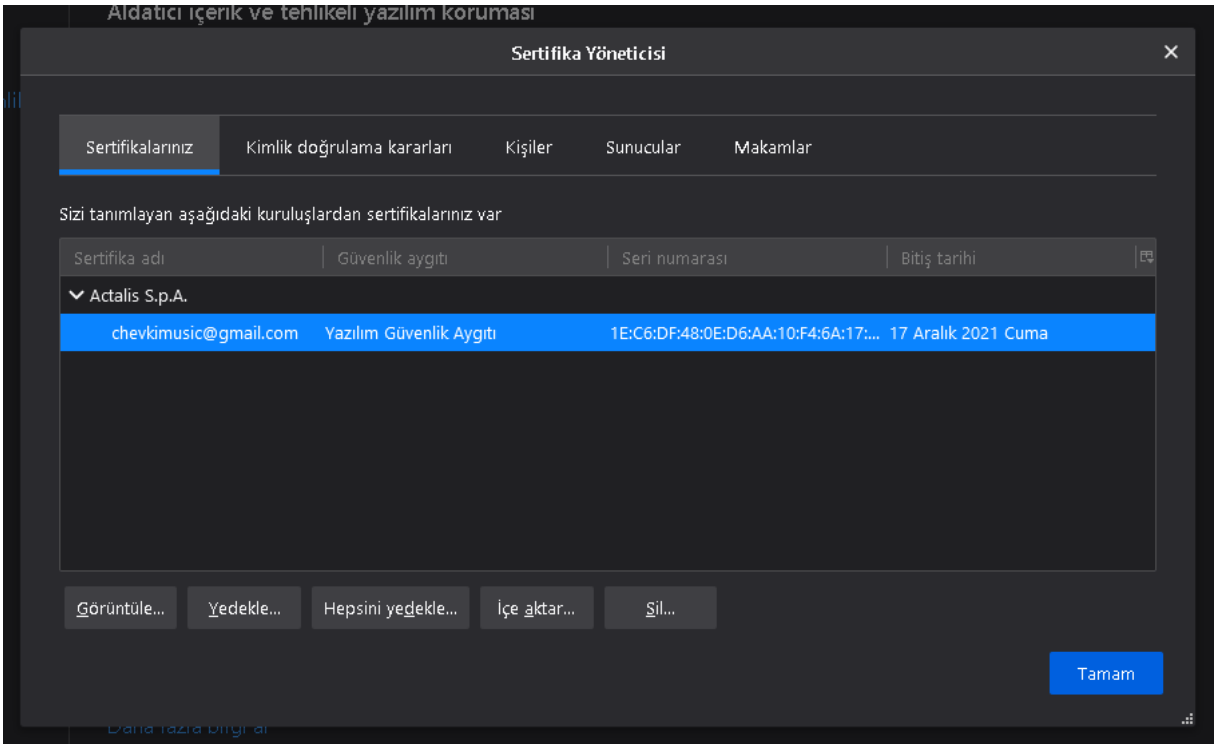
Exponent 65537

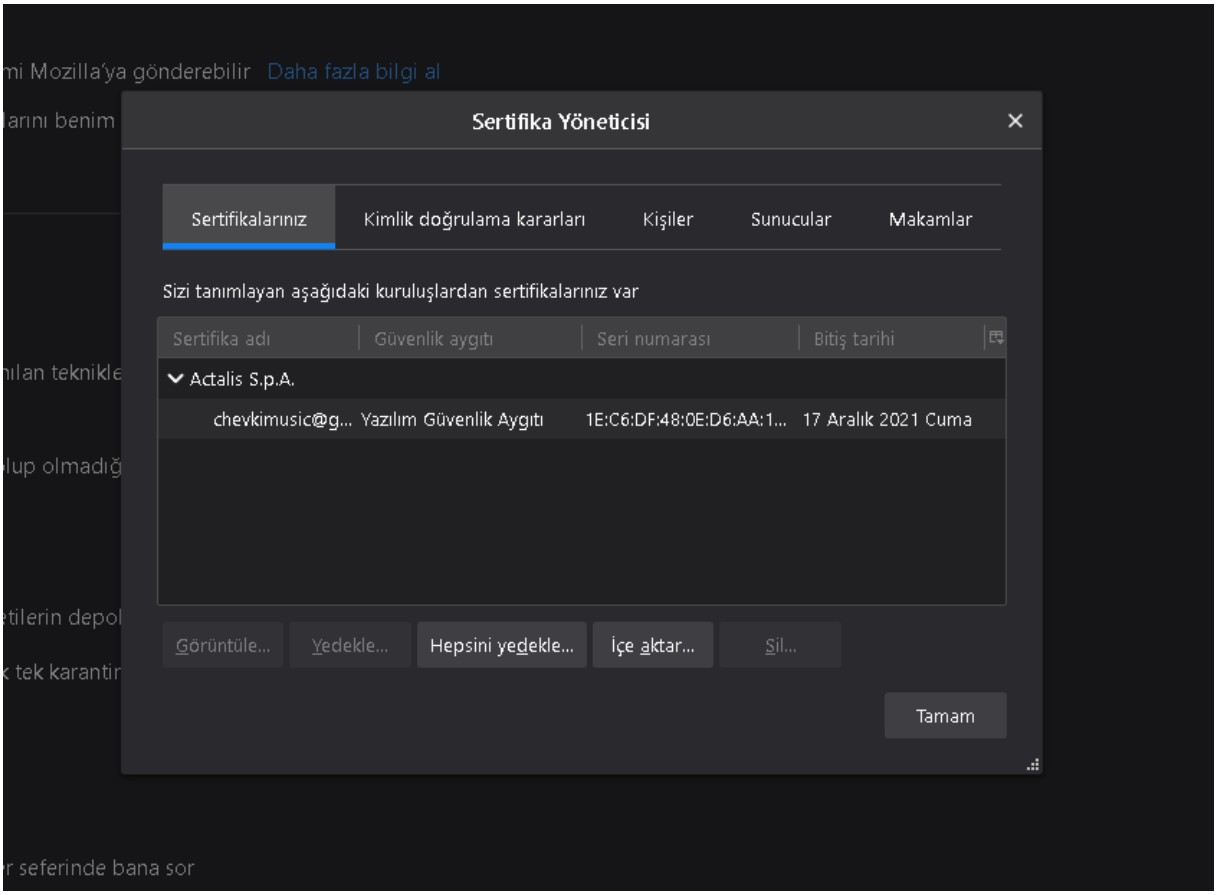
Modulus

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:





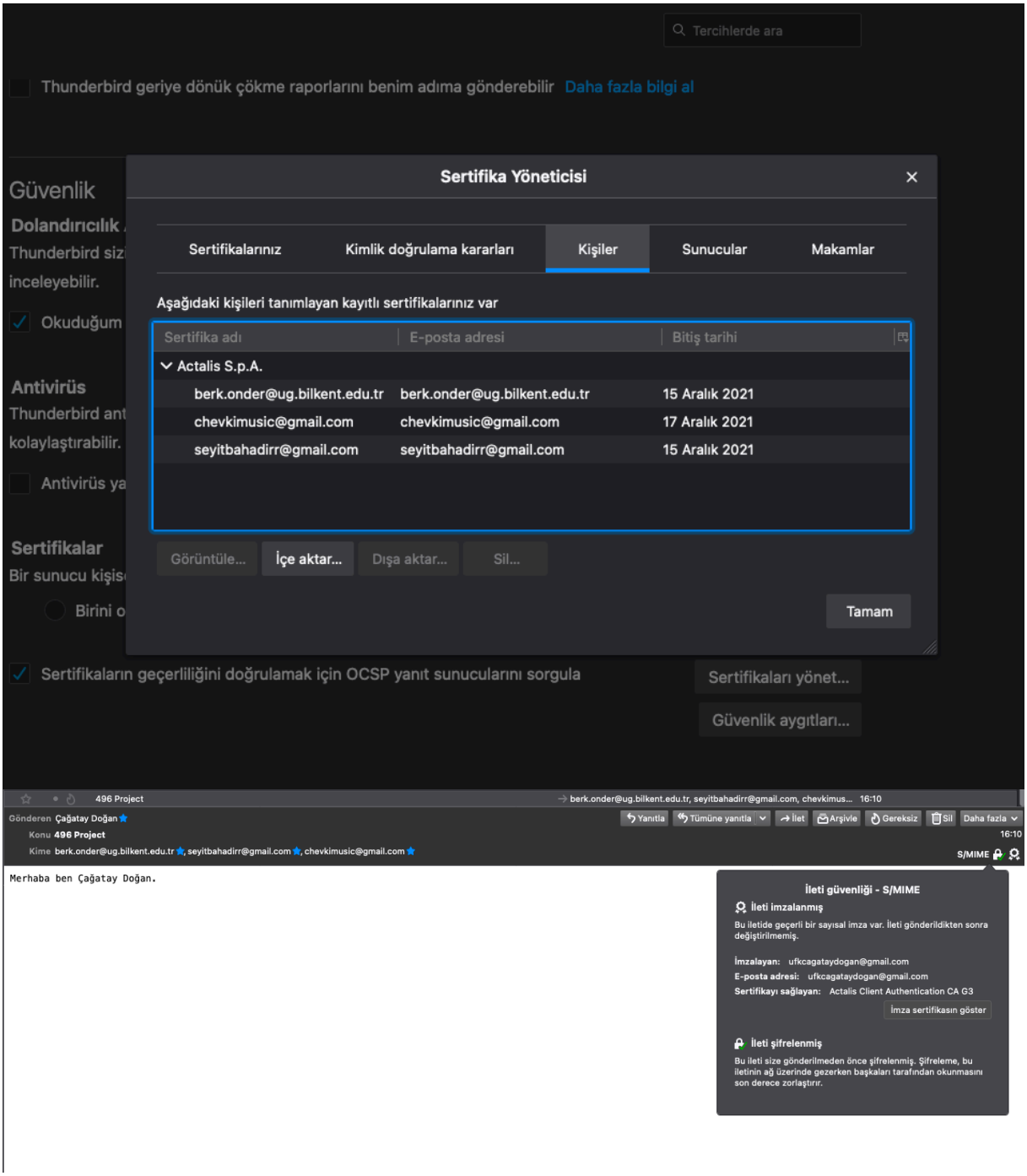


As we created our S/MIME certificates which are free from Actalis, then we can use them while we are sending encrypted e-mails via Mozilla Thunderbird. By doing this, our e-mails are secure, signed, and encrypted now. After we send e-mails to each other, we observed these e-mails cannot be seen on other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

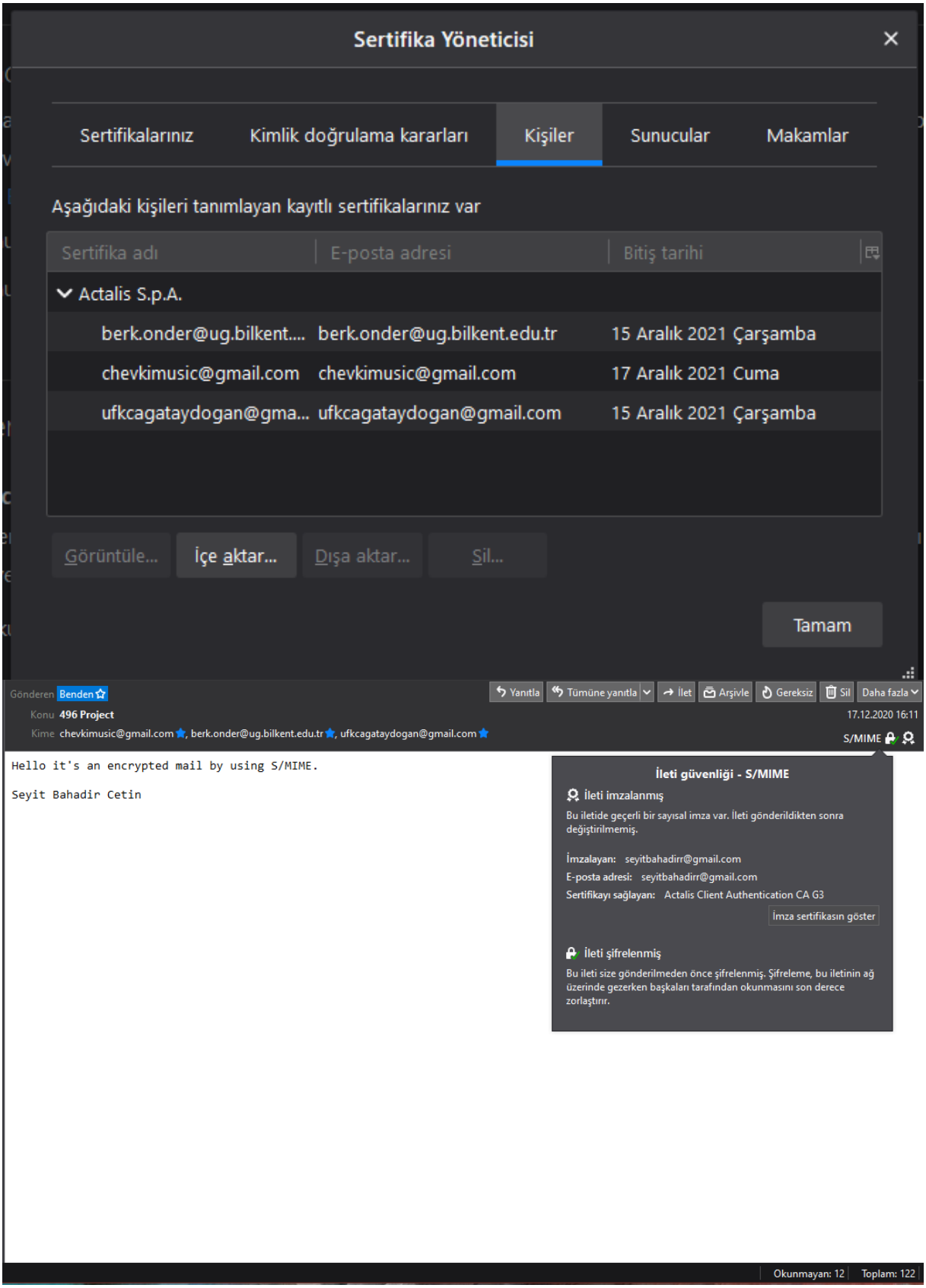
The details of the certificate issuer can be seen on Mozilla Thunderbird by doing following steps: Options-->Privacy&Security-->Certificates-->Manage Certificates-->Your Certificates-->Double Click on the Certificate-->Go to Issuer Name Part (Click on the Common Name). Now you can check the certificate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

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

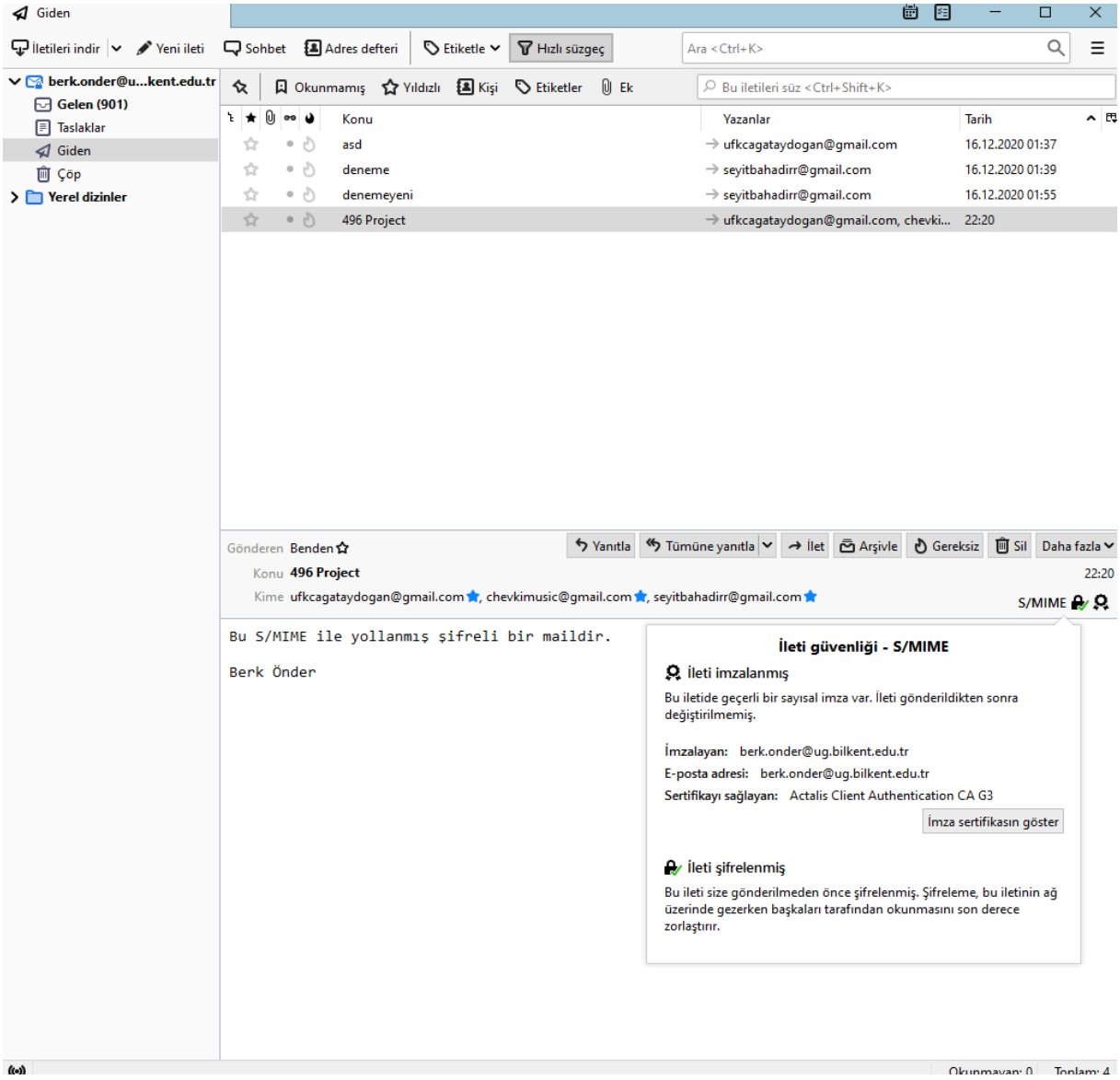
Çağatay DOĞAN Screenshots:

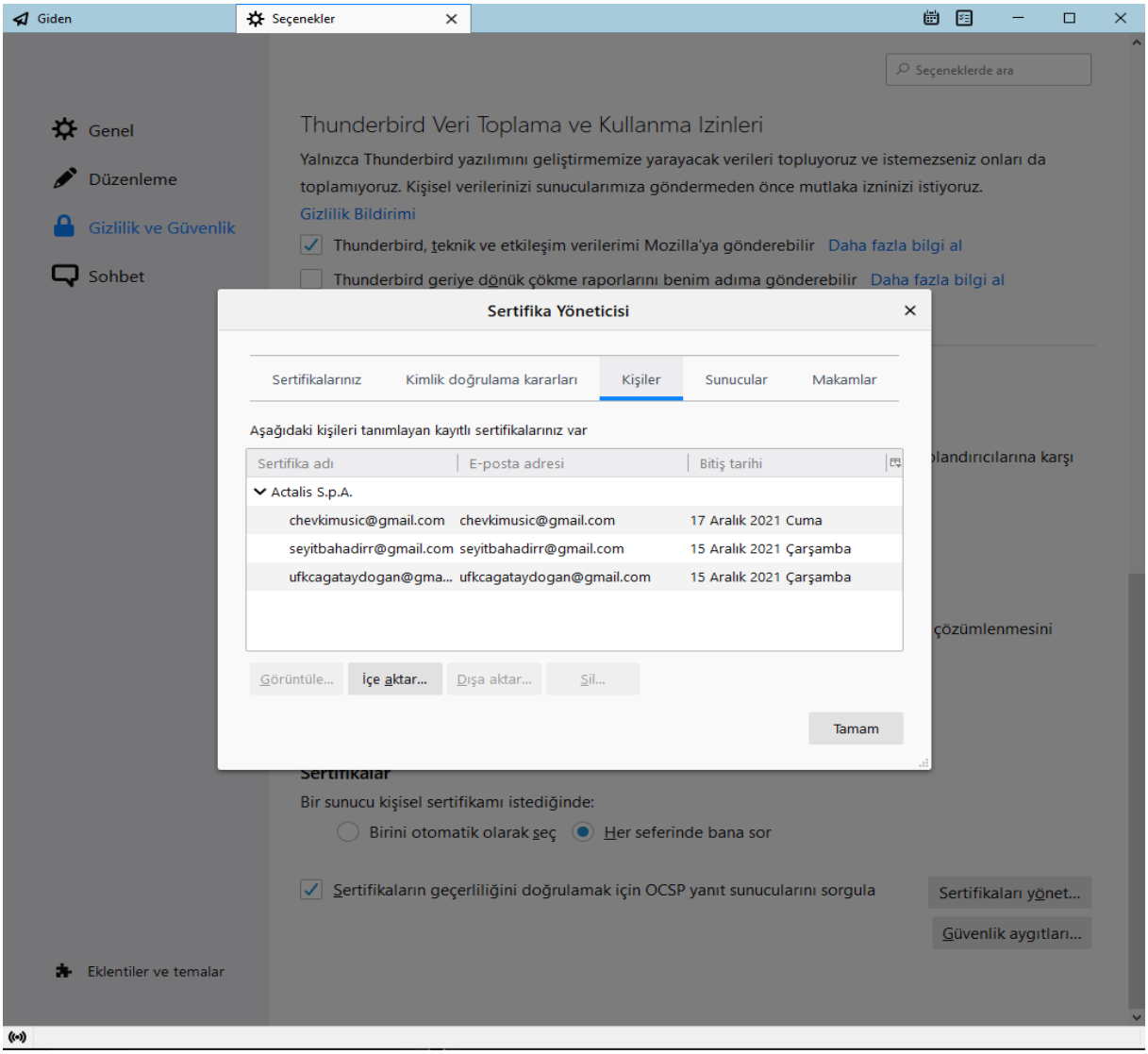


Bahadır ÇETİN Screenshots:

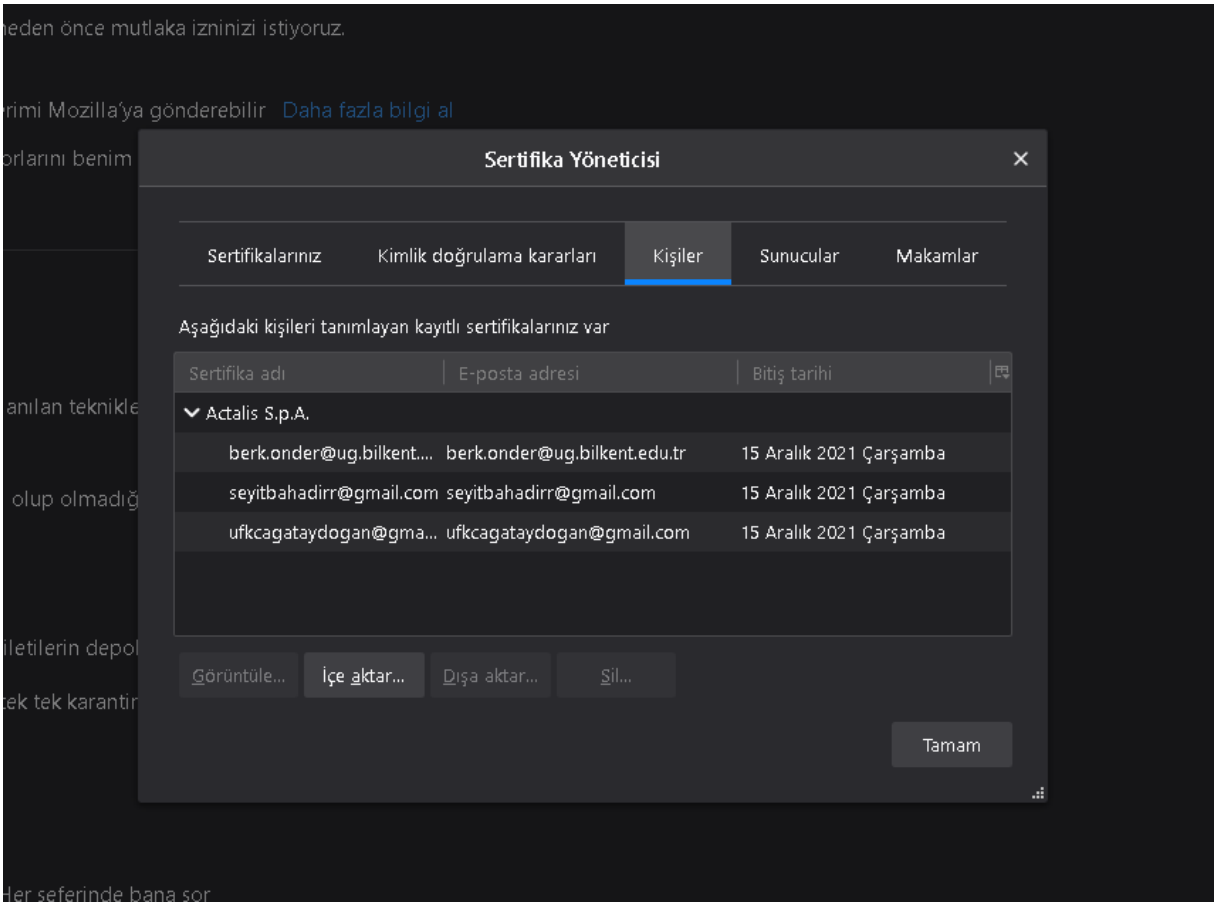


Berk ÖNDER Screenshots:



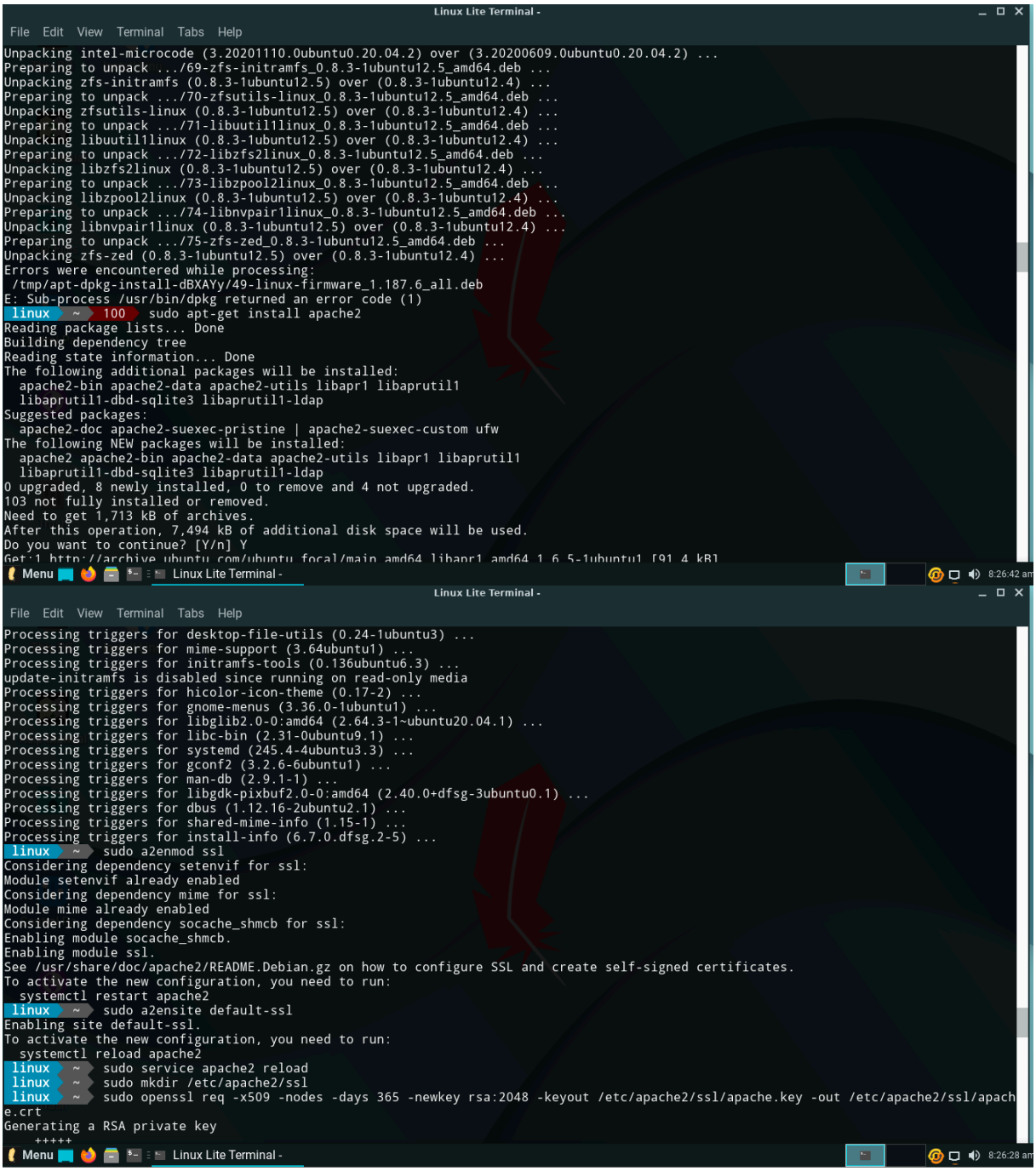

Giden
Seçenekler
Seçeneklerde ara
Genel
Düzenleme
Gizlilik ve Güvenlik
Sohbet
Thunderbird Veri Toplama ve Kullanma Izinleri
Yalnızca Thunderbird yazılımını geliştirmemize yarayacak verileri topluyoruz ve istemezseniz onları da toplamıyoruz. Kişisel verilerinizi sunucularımıza göndermeden önce mutlaka izninizi istiyoruz.
Gizlilik Bildirimi
Thunderbird, teknik ve etkileşim verilerimi Mozilla'ya gönderebilir Daha fazla bilgi al
Thunderbird geriye dönük çökme raporlarını benim adıma gönderebilir Daha fazla bilgi al
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.
chevkimusic@gmail.com chevkimusic@gmail.com 17 Aralık 2021 Cuma
seyitbahadirr@gmail.com seyitbahadirr@gmail.com 15 Aralık 2021 Çarşamba
ufkcagataydogan@gma... ufkcagataydogan@gmail.com 15 Aralık 2021 Çarşamba
Görüntüle...
İçe aktar...
Dışa aktar...
Sil...
Tamam
Bir sunucu kişisel sertifikamı istediğinde:
Birini otomatik olarak seç
Her seferinde bana sor
Sertifikaların geçerliliğini doğrulamak için OCSP yanıt sunucularını sorgula
Sertifikaları yönet...
Güvenlik aygıtları...
Eklentiler ve temalar

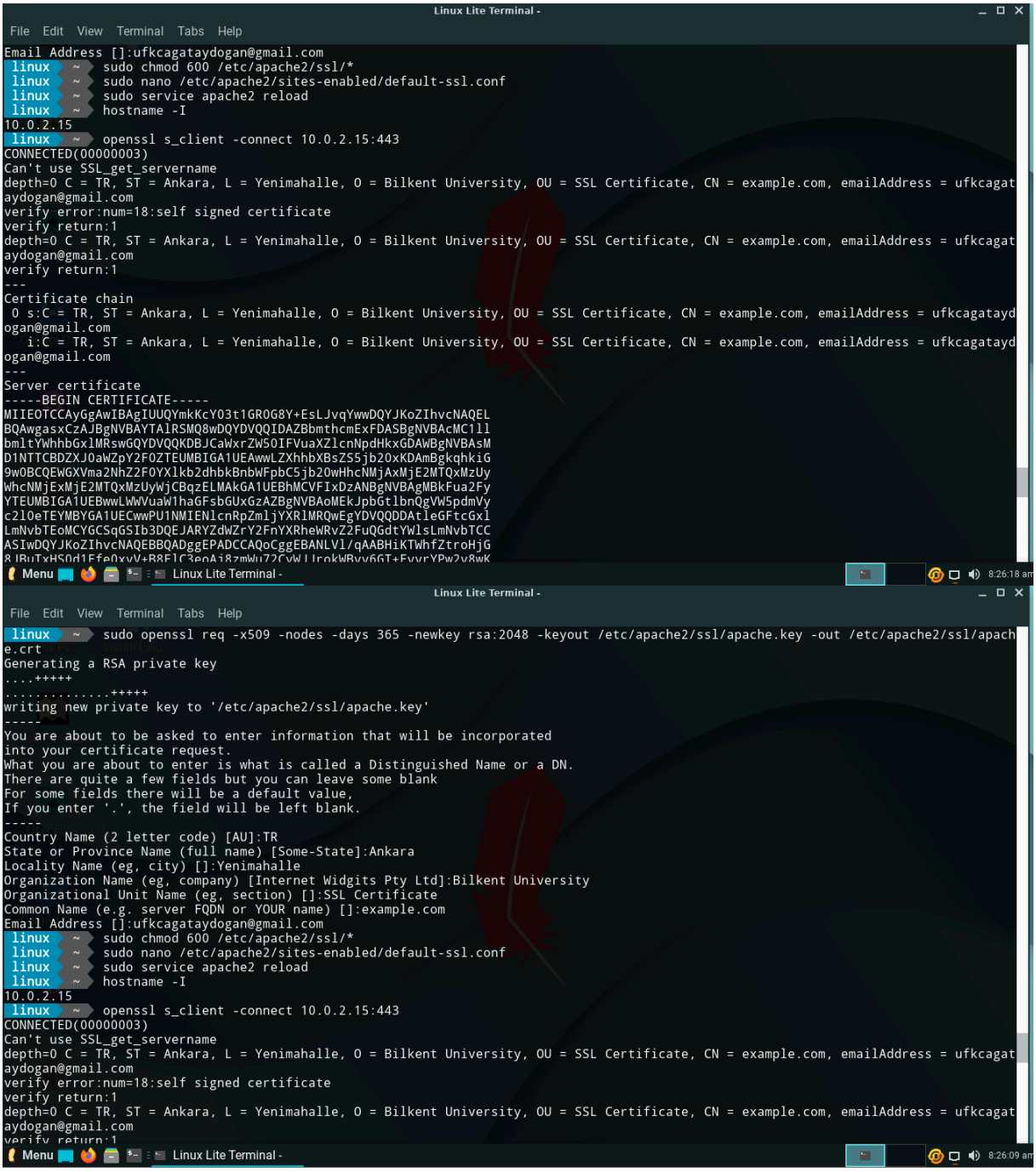
Armağan OĞUZ Screenshots:



3. Part of The Project: Configuring Apache Webserver

Çağatay DOĞAN Screenshots:

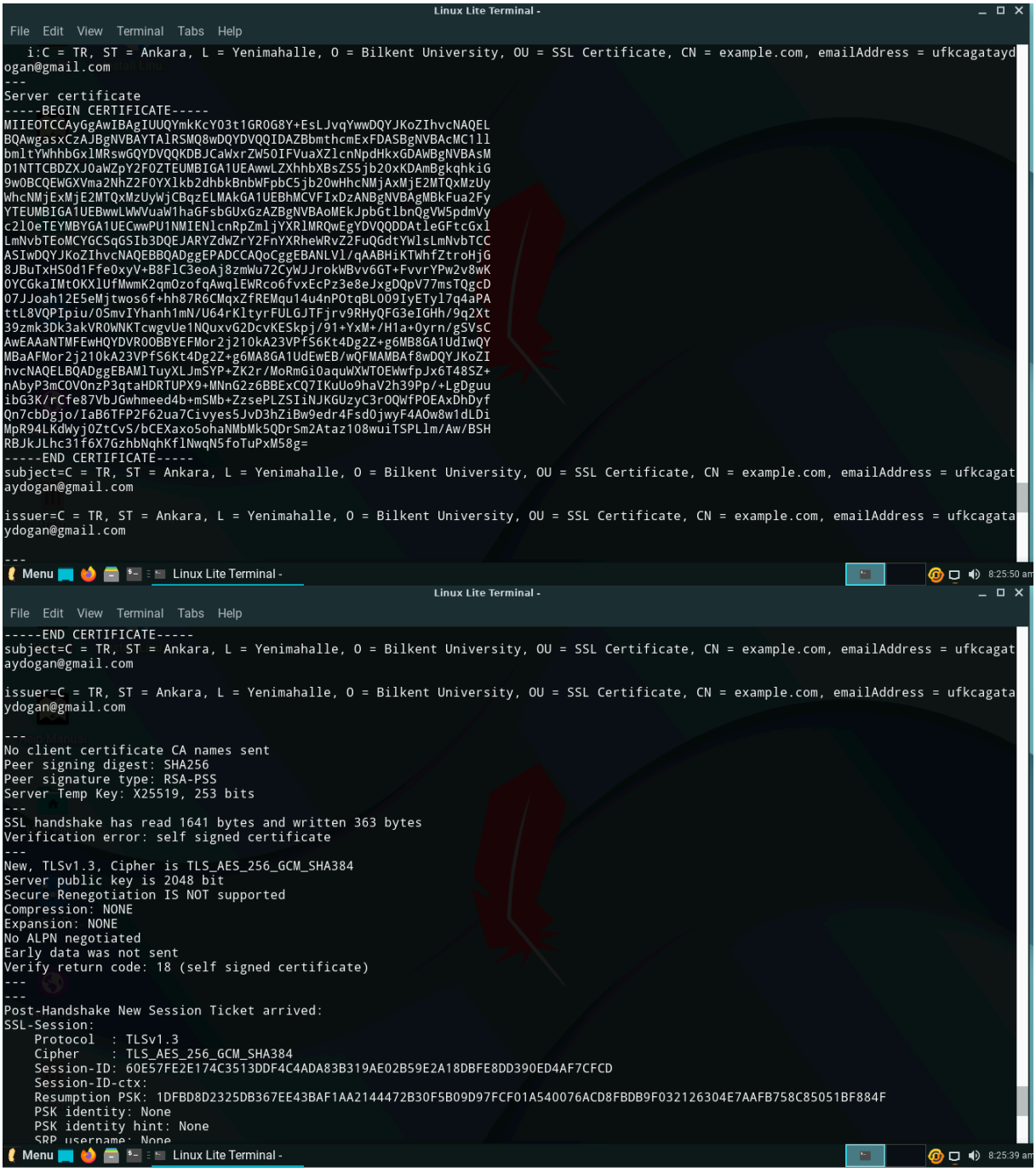



Linux Lite Terminal -
File Edit View Terminal Tabs Help
Email Address []:ufkcagataydogan@gmail.com
linux ~ sudo chmod 600 /etc/apache2/ssl/*
linux ~ sudo nano /etc/apache2/sites-enabled/default-ssl.conf
linux ~ sudo service apache2 reload
linux ~ hostname -I
10.0.2.15
linux ~ openssl s_client -connect 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-----
MIIEOTCCAyGgAwIBAgIUUQYmkKcY03t1GROG8Y+EsLJvqYwwDQYJKoZIhvcNAQEL
BQAwgasxCzAJBgNVBAYTAlRSMQ8wDQYDVQQIDAZBbmthcmExFDASBgNVBAcMC1ll
bmltYWhhbGxlMRswGQYDVQQKDBJCaWxrZW50IFVuaXZlcnNpdHkxGDAWBgNVBAsM
D1NTTCBDZXJ0aWZpY2F0ZTEUMBIGA1UEAwwLZXhhbXBsZS5jb20xKDAmBgkqhkiG
9w0BCQEWGXVma2NhZ2FOYXlkb2dhbkBnbWFpbC5jb20wHhcNMjAxMjE2MTQxMzUy
WhcNMjExMjE2MTQxMzUyWjCBqzELMAkGA1UEBhMCVFIxDzANBgNVBAgMBkFua2Fy
YTEUMBIGA1UEBwwLWWVuaW1haGFsbGUxGzAZBgNVBAoMEkJpbGtlbnQgVW5pdmVy
c2l0eTEYMBYGA1UECwwPU1NMIENlcnRpZmljYXRlMRQwEgYDVQQDDAtleGFtcGxl
LmNvbTEoMCYGCSqGSIb3DQEJARYZdWZrY2FnYXRheWRvZ2FuQGdtYWlsLmNvbTCC
ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANLVl/qAABHiKTWhfZtroHjG
8JBuTxHSOd1Ffe0xyV+R8FlC3eoAi8zmWu72CvWJJrokWRyv6GT+FyyrYPw2v8wK
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 = ufkcagatayd
ogan@gmail.com

Server certificate
-----BEGIN CERTIFICATE-----
MIIEOTCCAyGgAwIBAgIUUQYmkKcYO3t1GROG8Y+EsLJvqYwwDQYJKoZIhvcNAQEL
BQAwgasxCzAJBgNVBAYTAlRSMQ8wDQYDVQQIDAZBbmthcmExFDASBgNVBAcMC1ll
bmltYWhhbGxlMRswGQYDVQQKDBJCaWxrZW50IFVuaXZlcnNpdHkxGDAWBgNVBAsM
D1NTTCBDZXJ0aWZpY2F0ZTEUMBIGA1UEAwwLZXhhbXBsZS5jb20xKDAmBgkqhkiG
9w0BCQEWGXVma2NhZ2FOYXlkb2dhbkBnbWFpbC5jb20wHhcNMjAxMjE2MTQxMzUy
WhcNMjExMjE2MTQxMzUyWjCBqzELMAkGA1UEBhMCVFIxDzANBgNVBAgMBkFua2Fy
YTEUMBIGA1UEBwwLWWVuaW1haGFsbGUxGzAZBgNVBAoMEkJpbGtlbnQgVW5pdmVy
c2l0eTEYMBYGA1UECwwPU1NMIENlcnRpZmljYXRlMRQwEgYDVQQDDAtleGFtcGxl
LmNvbTEoMCYGCSqGSIb3DQEJARYZdWZrY2FnYXRheWRvZ2FuQGdtYWlsLmNvbTCC
ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANLVl/qAABHiKTWhfZtroHjG
8JBuTxHSOd1Ffe0xyV+B8FlC3eoAj8zmWu72CyWJJrokWBvv6GT+FvvrYPw2v8wK
OYCGkaIMtOKXlUfMwmK2qmOzofqAwqlEWRco6fvxEcPz3e8eJxgDQpV77msTQgcD
O7JJoah12E5eMjtwos6f+hh87R6CMqxZfREMqu14u4nPOtqBLOO9IyETy17q4aPA
ttL8VQPIpiu/OSmvIYhanh1mN/U64rKltyrFULGJTFjrv9RHyQFG3eIGHh/9q2Xt
39zmk3Dk3akVROWNKTcwgvUe1NQuxvG2DcvKESkpj/91+YxM+/H1a+Oyrn/gSVsC
AwEAAaNTMFEwHQYDVR0OBBYEFMor2j21OkA23VPfS6Kt4Dg2Z+g6MB8GA1UdIwQY
MBaAFMor2j21OkA23VPfS6Kt4Dg2Z+g6MA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZI
hvcNAQELBQADggEBAMlTuyXLJmSYP+ZK2r/MoRmGiOaquWXWTOEWwfpJx6T48SZ+
nAbyP3mCOVOnzP3qtaHDRTUPX9+MNnG2z6BBExCQ7IKuUo9haV2h39Pp/+LgDguu
ibG3K/rCfe87VbJGwhmeed4b+mSMb+ZzsePLZSIiNJKGUzyC3rOQWfPOEAxDhDyf
Qn7cbDgjo/IaB6TFP2F62ua7Civyes5JvD3hZiBw9edr4FsdOjwyF4AOw8w1dLDi
MpR94LKdWyjOZtCvS/bCEXaxo5ohaNMbMk5QDrSm2Ataz1O8wuiTSPLlm/Aw/BSH
RBJkJLhc31f6X7GzhbNqhKflNwqN5foTuPxM58g=
-----END CERTIFICATE-----
subject=C = TR, ST = Ankara, L = Yenimahalle, O = Bilkent University, OU = SSL Certificate, CN = example.com, emailAddress = ufkcagat
aydogan@gmail.com
issuer=C = TR, ST = Ankara, L = Yenimahalle, O = Bilkent University, OU = SSL Certificate, CN = example.com, emailAddress = ufkcagata
ydogan@gmail.com

Menu
Linux Lite Terminal -
8:25:50 am
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 = ufkcagat
aydogan@gmail.com
issuer=C = TR, ST = Ankara, L = Yenimahalle, O = Bilkent University, OU = SSL Certificate, CN = example.com, emailAddress = ufkcagata
ydogan@gmail.com

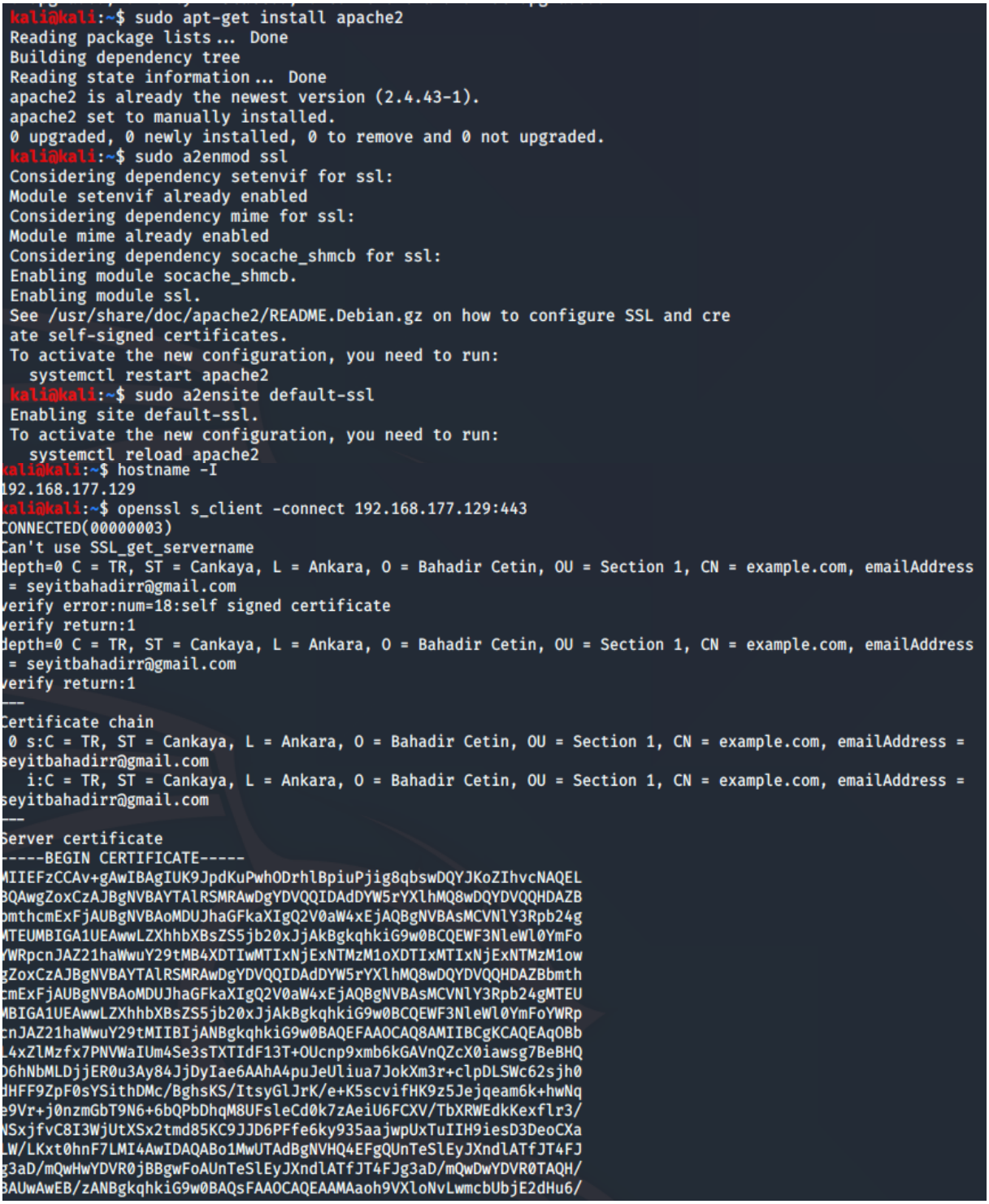
No client certificate CA names sent
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits

SSL handshake has read 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)

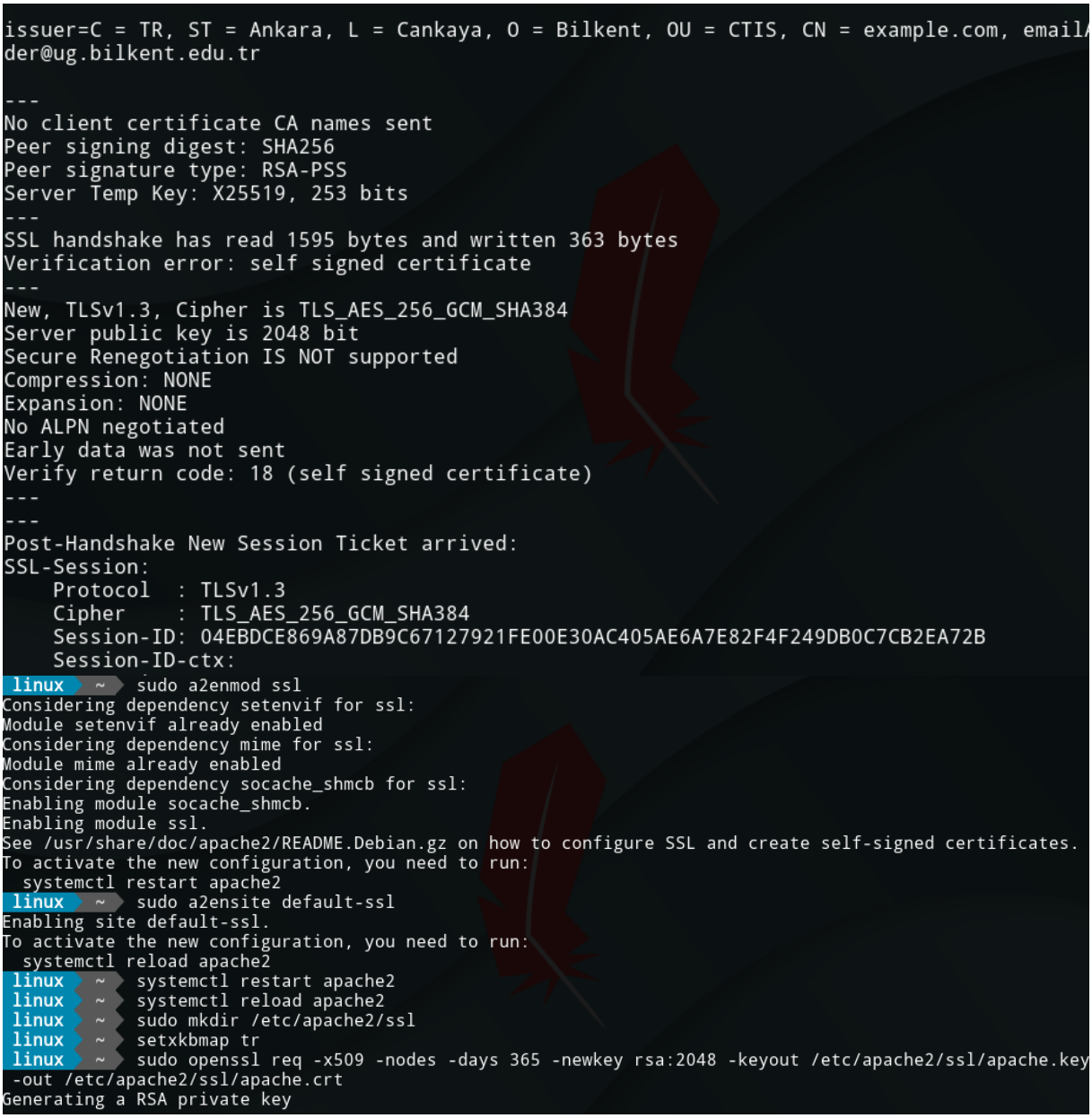
Post-Handshake New Session Ticket arrived:
SSL-Session:
Protocol : TLSv1.3
Cipher : TLS_AES_256_GCM_SHA384
Session-ID: 60E57FE2E174C3513DDF4C4ADA83B319AE02B59E2A18DBFE8DD390ED4AF7CFCD
Session-ID-ctx:
Resumption PSK: 1DFBD8D2325DB367EE43BAF1AA2144472B30F5B09D97FCF01A540076ACD8FBDB9F032126304E7AAFB758C85051BF884F
PSK identity: None
PSK identity hint: None
SRP username: None
Menu
Linux Lite Terminal -
8:25:39 am

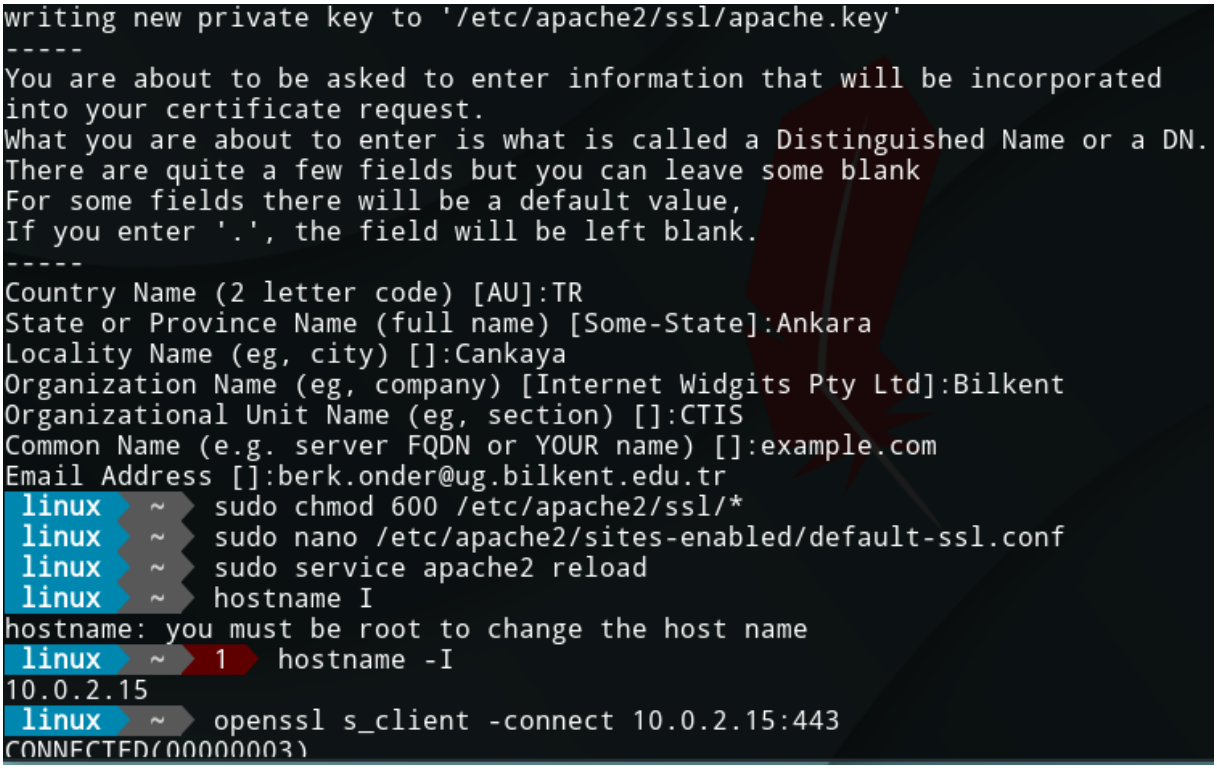
Bahadır ÇETİN Screenshots:



```
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/ss
l/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 []:seyitbahadirr@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 192.168.177.129:443
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
 = seyitbahadirr@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
 = seyitbahadirr@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 =
seyitbahadirr@gmail.com
   i:C = TR, ST = Cankaya, L = Ankara, O = Bahadir Cetin, OU = Section 1, CN = example.com, emailAddress =
seyitbahadirr@gmail.com
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIEFzCCAv+gAwIBAgIUK9JpdKuPwhODrhlBpiuPjig8qbswDQYJKoZIhvcNAQEL
BQAwgZoxCzAJBgNVBAYTAlRSMRAwDgYDVQQIDAdDYW5rYXlhMQ8wDQYDVQQHDAZB
bmthcmExFjAUBgNVBAoMDUJhaGFkaXIgQ2V0aW4xEjAQBgNVBAsMCVNlY3Rpb24g
MTEUMBIGA1UEAwwLZXhhbXBsZS5jb20xJjAkBgkqhkiG9w0BCQEWF3NleWl0YmFo
YWRpcnJAZ21haWwuY29tMB4XDTIwMTIxNjExNTMzM1oXDTIxMTIxNjExNTMzM1ow
gZoxCzAJBgNVBAYTAlRSMRAwDgYDVQQIDAdDYW5rYXlhMQ8wDQYDVQQHDAZBbmth
cmExFjAUBgNVBAoMDUJhaGFkaXIgQ2V0aW4xEjAQBgNVBAsMCVNlY3Rpb24gMTEU
MBIGA1UEAwwLZXhhbXBsZS5jb20xJjAkBgkqhkiG9w0BCQEWF3NleWl0YmFoYWRp
cnJAZ21haWwuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAqOBb
L4xZlMzfx7PNVWaIUm4Se3sTXTIdF13T+OUcnp9xmb6kGAVnQZcX0iawsg7BeBHQ
D6hNbMLDjjER0u3Ay84JjDyIae6AAhA4puJeUliua7JokXm3r+clpDLSWc62sjh0
dHFF9ZpF0sYSithDMc/BghsKS/ItsyGlJrK/e+K5scvifHK9z5Jejqeam6k+hwNq
e9Vr+j0nzmGbT9N6+6bQPbDhqM8UFsleCd0k7zAeiU6FCXV/TbXRWEdkKexflr3/
NSxjfvC8I3WjUtXSx2tmd85KC9JJD6PFfe6ky935aajwpUxTuIIH9iesD3DeoCXa
LW/LKxt0hnF7LMI4AwIDAQABo1MwUTAdBgNVHQ4EFgQUnTeSlEyJXndlATfJT4FJ
g3aD/mQwHwYDVR0jBBgwFoAUnTeSlEyJXndlATfJT4FJg3aD/mQwDwYDVR0TAQH/
BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEAAMAaoh9VXloNvLwmcbUbjE2dHu6/
/Y2OAcxKvvkTvWfO8jCB3ctFv587Czy2uK4fWnSAmMIMakd8LSDU0f016Vwenghz
kwad9qGVVrTcfSNsYorajMojKgaOP5yNl5P4+LMPqC2PfipgnNk3gXVJqVc+L4Me
jPW6iHCttqLP5mGfZpmDdO2Fs2ohNmhPUSyDYj/5XJR8IU7txbtHEjO5AyjoHqmm
gZ96KDnfeD+9N5shzzqswE+4kNus55T8Z3G+OymYNnDgYct2UJnrYVb4hnTzkqNU
DN71WO4LAbHvrMJy/fgXI0uWnH4s5J28jiTYRUd2pEeWrP0/ApH0C4oQag==
-----END CERTIFICATE-----
subject=C = TR, ST = Cankaya, L = Ankara, O = Bahadir Cetin, OU = Section 1, CN = example.com, emailAddress
 = seyitbahadirr@gmail.com

issuer=C = TR, ST = Cankaya, L = Ankara, O = Bahadir Cetin, OU = Section 1, CN = example.com, emailAddress
= seyitbahadirr@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 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:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: F273F4D08753CAB15DEE120D7B6D400F940DE89F13C9C2C5CE31802E6AA5FF92
    Session-ID-ctx:
    Resumption PSK: F17B114901BDD309A8D525CD5186E3811B8A666F37D8468DC778C2E662F3DA561D621FD609DB460FDA12352
D2EBB6B7D
    PSK identity: None
    PSK identity hint: None
```

Berk ÖNDER Screenshots:

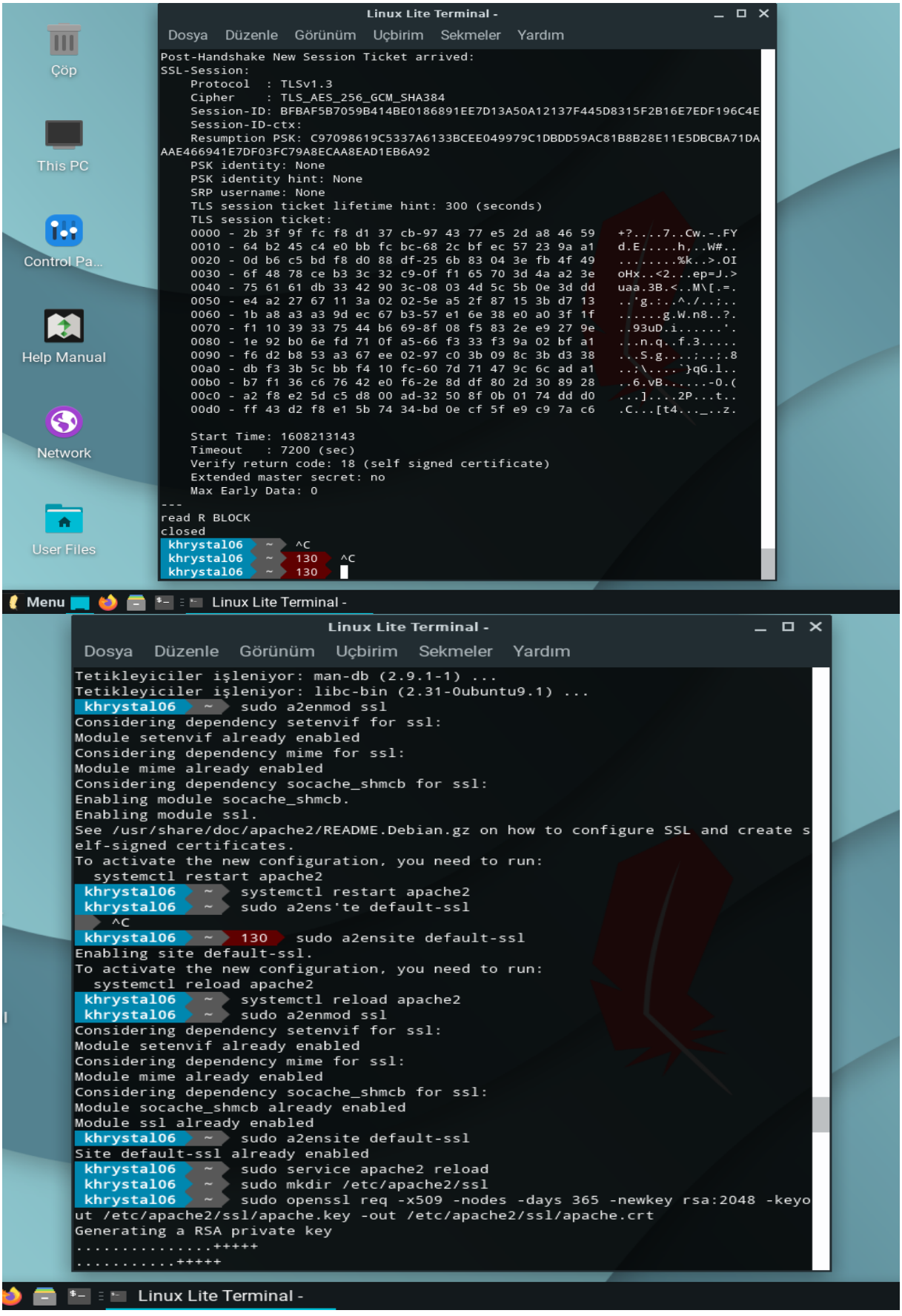


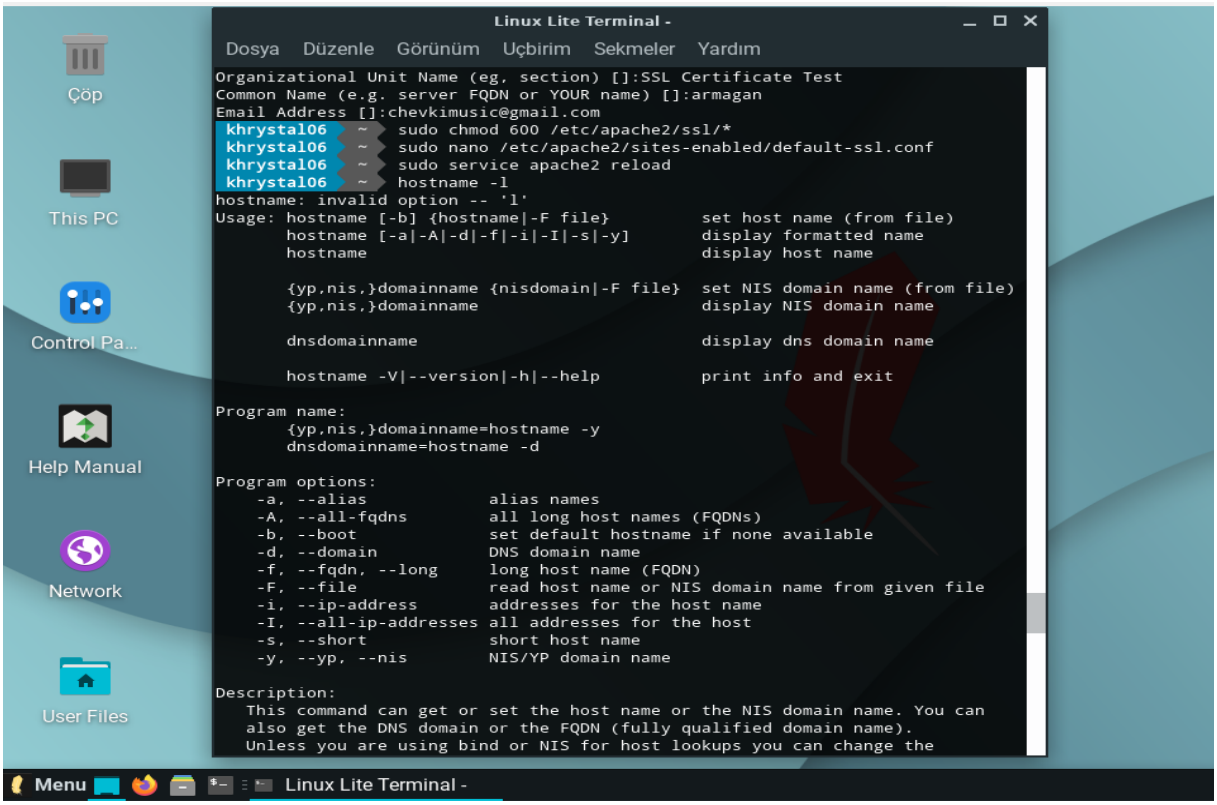

writing new private key to '/etc/apache2/ssl/apache.key'

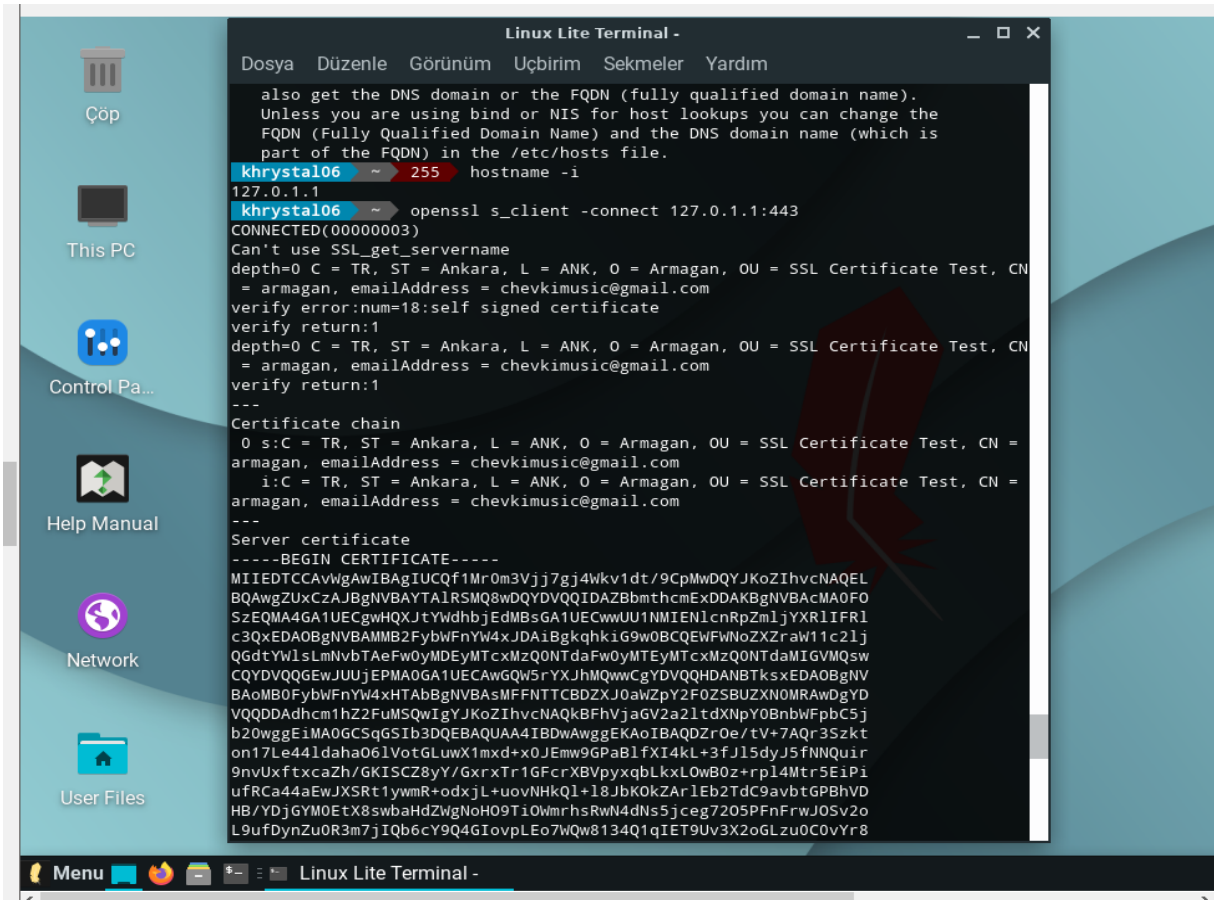
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', 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:

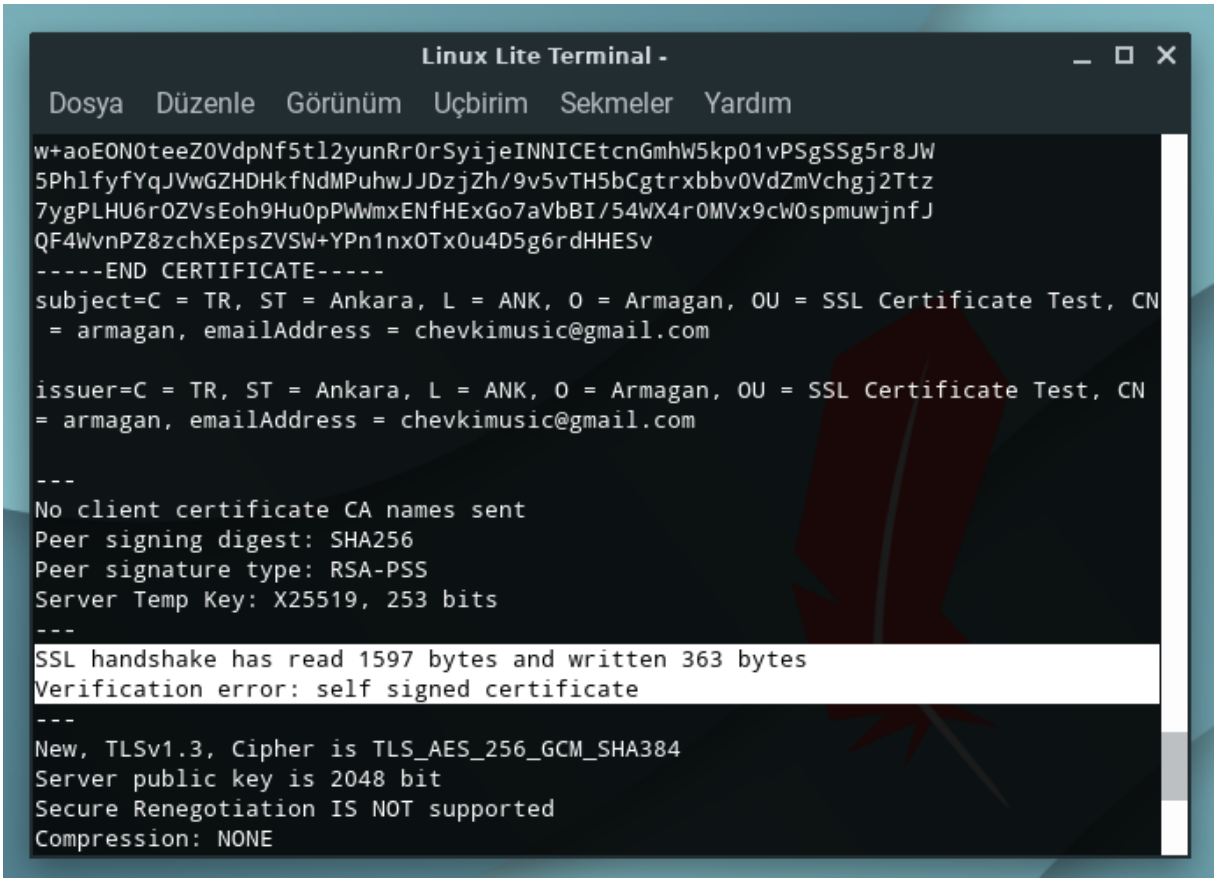



Linux Lite Terminal -
Dosya Düzenle Görünüm Uçbirim Sekmeler Yardım
Organizational Unit Name (eg, section) []:SSL Certificate Test
Common Name (e.g. server FQDN or YOUR name) []:armagan
Email Address []:chevkimusic@gmail.com
khrystal06 ~ sudo chmod 600 /etc/apache2/ssl/*
khrystal06 ~ sudo nano /etc/apache2/sites-enabled/default-ssl.conf
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
Çöp
This PC
Control Pa...
Help Manual
Network
User Files
Menu
Linux Lite Terminal -


Linux Lite Terminal -
Dosya Düzenle Görünüm Uçbirim Sekmeler Yardım
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-----
MIIEDTCCAvWgAwIBAgIUCQf1Mr0m3Vjj7gj4Wkv1dt/9CpMwDQYJKoZIhvcNAQEL
BQAwgZUxCzAJBgNVBAYTAlRSMQ8wDQYDVQQIDAZBbmthcmExDDAKBgNVBAcMA0FO
SzEQMA4GA1UECgwHQXJtYWdhbjEdMBsGA1UECwwUU1NMIENlcnRpZmljYXRlIFRl
c3QxEDAOBgNVBAMMB2FybWFnYW4xJDAiBgkqhkiG9w0BCQEWFWNoZXZraW11c21j
QGdtYWlsLmNvbTAeFw0yMDEyMTcxMzQ0NTdaFw0yMTEyMTcxMzQ0NTdaMIGVMQsw
CQYDVQQGEwJUUjEPMA0GA1UECAwGQW5rYXJhMQwwCgYDVQQHDANBTksxEDAOBgNV
BAoMB0FybWFnYW4xHTAbBgNVBAsMFFNTTCBDZXJ0aWZpY2F0ZSBUZXN0MRAwDgYD
VQQDDAdhcm1hZ2FuMSQwIgYJKoZIhvcNAQkBFhVjaGV2a2ltdXNpY0BnbWFpbC5j
b20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDZr0e/tV+7AQr3Szkt
on17Le44ldahaO6lVotGLuwX1mxd+x0JEmw9GPaBlfXI4kL+3fJl5dyJ5fNNQuir
9nvUxftxcaZh/GKISCZ8yY/GxrxTr1GFcrXBVpyxqbLkxLOwB0z+rpl4Mtr5EiPi
ufRCa44aEwJXSRt1ywmR+odxjL+uovNHkQl+l8JbKOkZArlEb2TdC9avbtGPBhVD
HB/YDjGYM0EtX8swbaHdZWgNoHO9TiOWmrhsRwN4dNs5jceg72O5PFnFrwJOSv2o
L9ufDynZu0R3m7jIQb6cY9Q4GIovpLEo7WQw8134Q1qIET9Uv3X2oGLzu0C0vYr8
Çöp
This PC
Control Pa...
Help Manual
Network
User Files
Menu
Linux Lite Terminal -


Linux Lite Terminal -
Dosya Düzenle Görünüm Uçbirim Sekmeler Yardım
w+aoEON0teeZ0VdpNf5tl2yunRr0rSyijeINNICEtcnGmhW5kp01vPSgSSg5r8JW
5PhlfyfYqJVwGZHDHkfNdMPuhwJJDzjZh/9v5vTH5bCgtrxbbv0VdZmVchgj2Ttz
7ygPLHU6rOZVsEoh9Hu0pPWWmxENfHExGo7aVbBI/54WX4r0MVx9cW0spmuwjnfJ
QF4WvnPZ8zchXEpsZVSW+YPn1nxOTx0u4D5g6rdHHESv
-----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

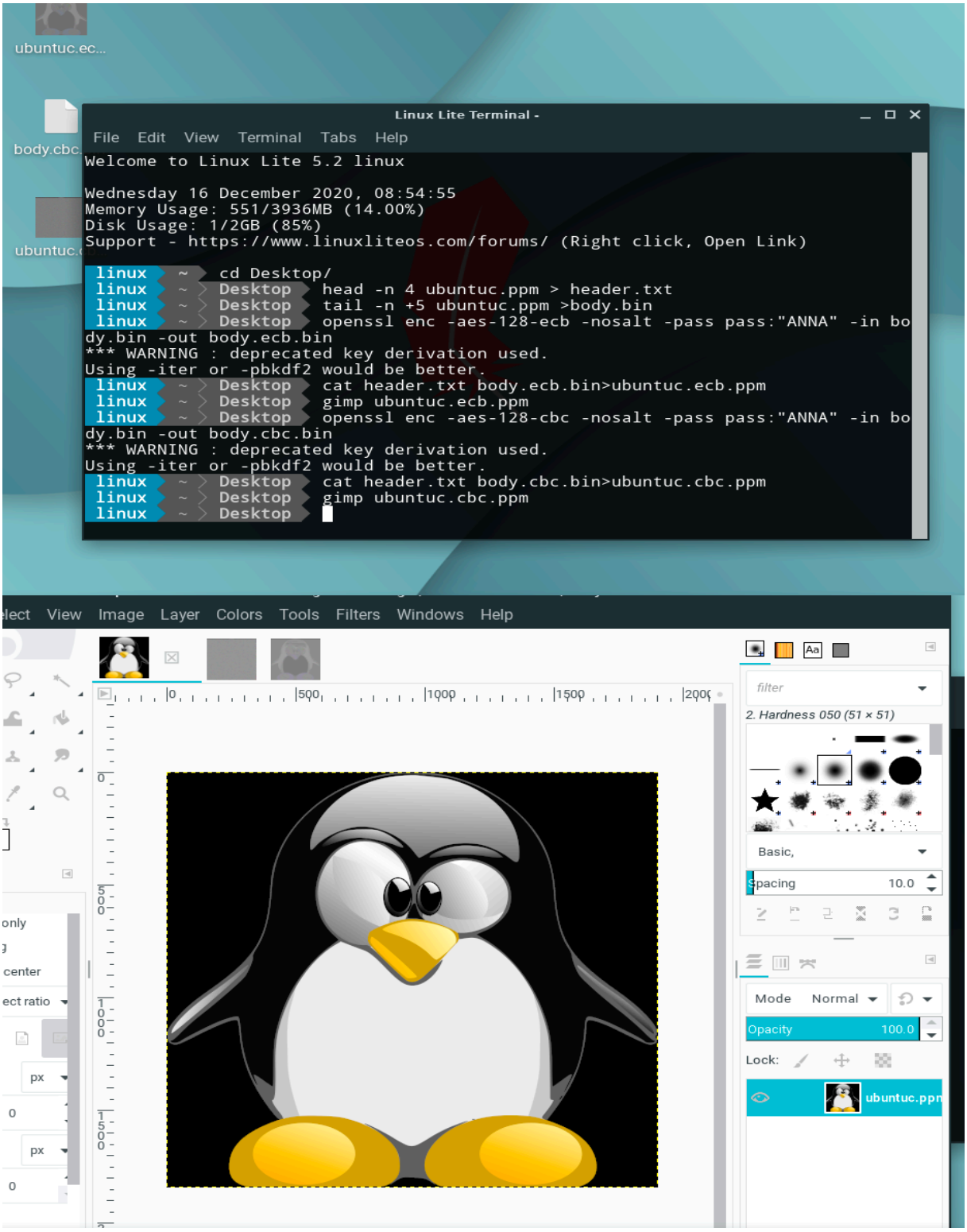
SSL handshake has read 1597 bytes and written 363 bytes
Verification error: self signed certificate

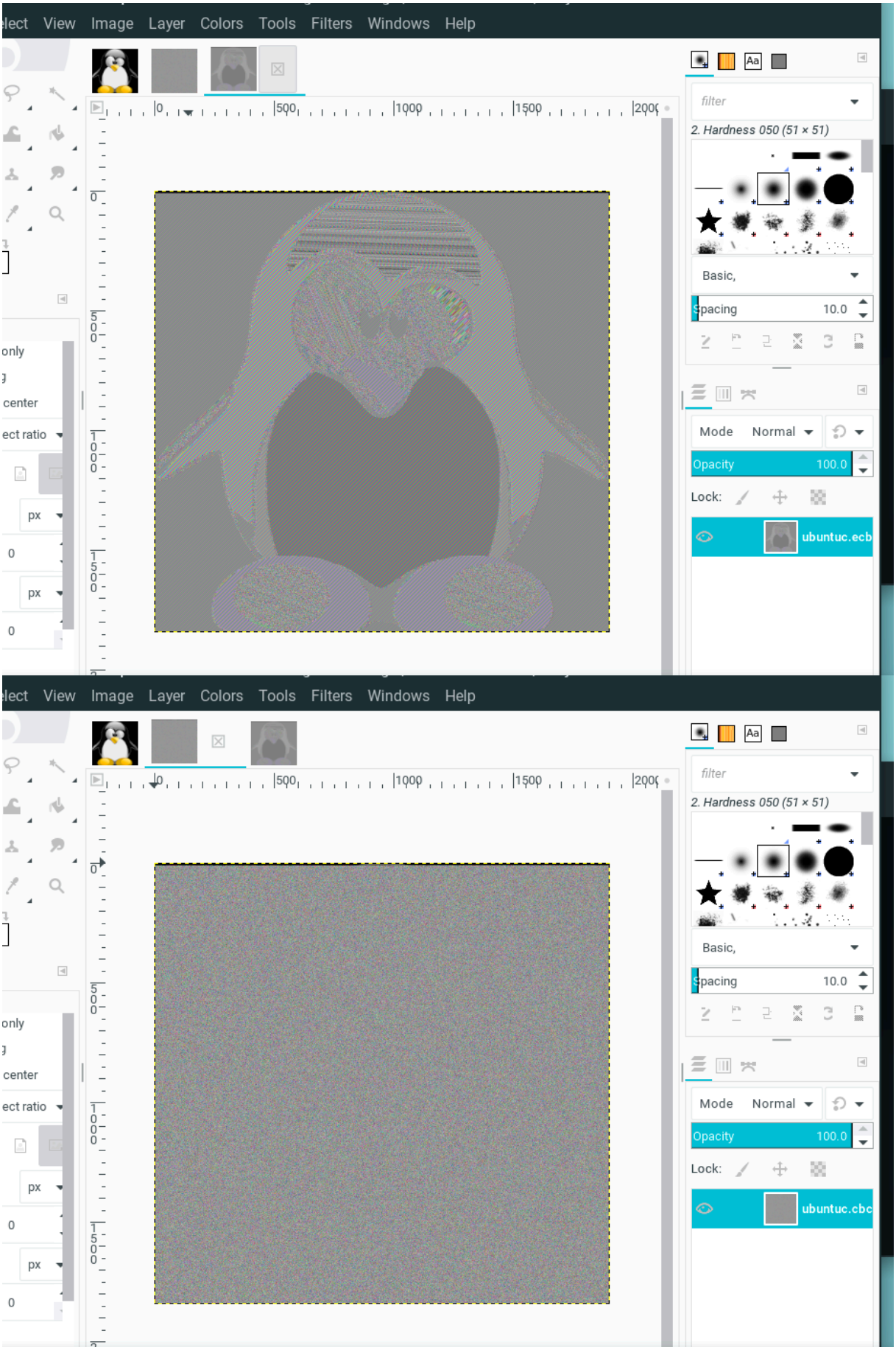
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE

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

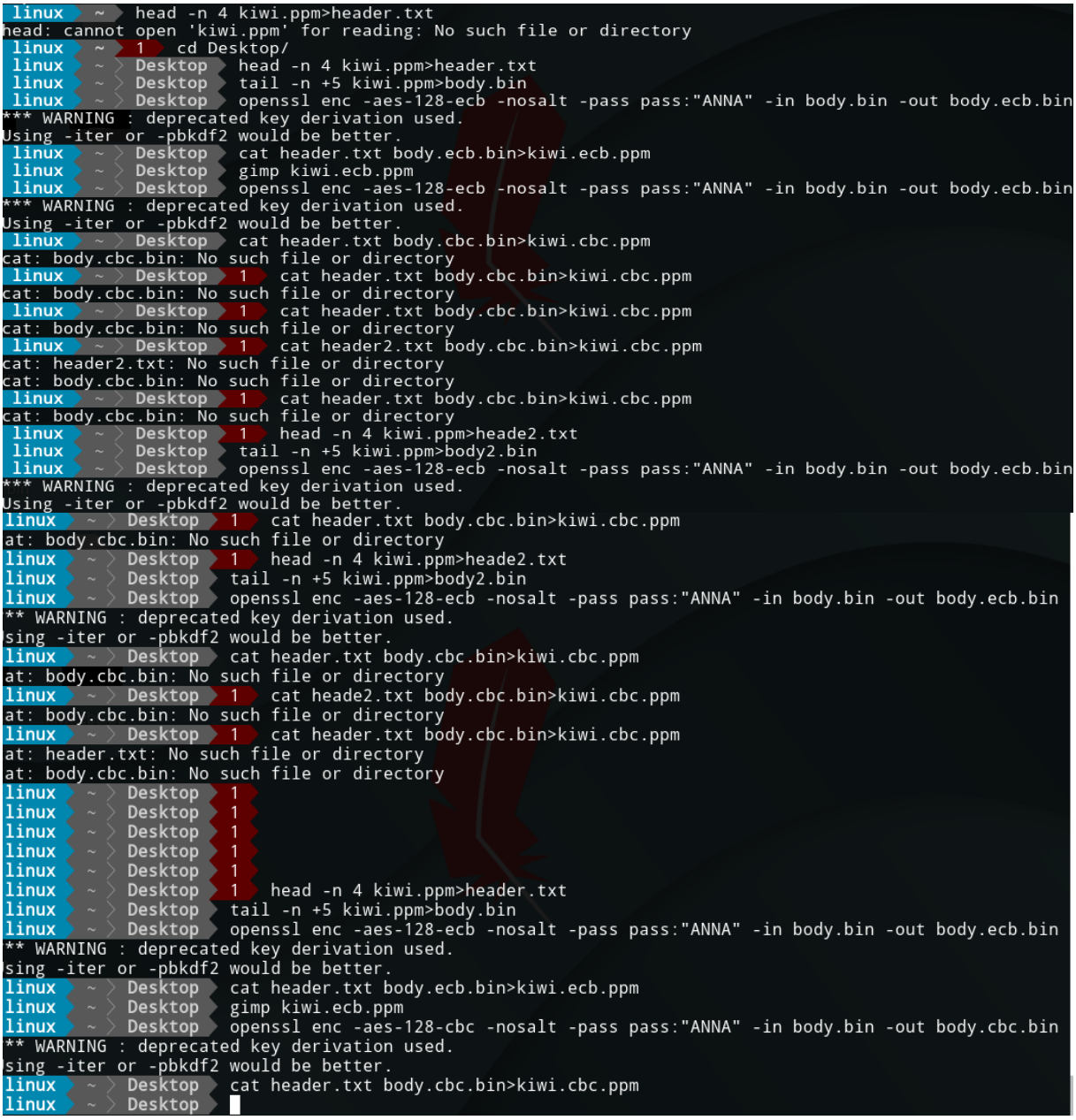
Encrypting an Image

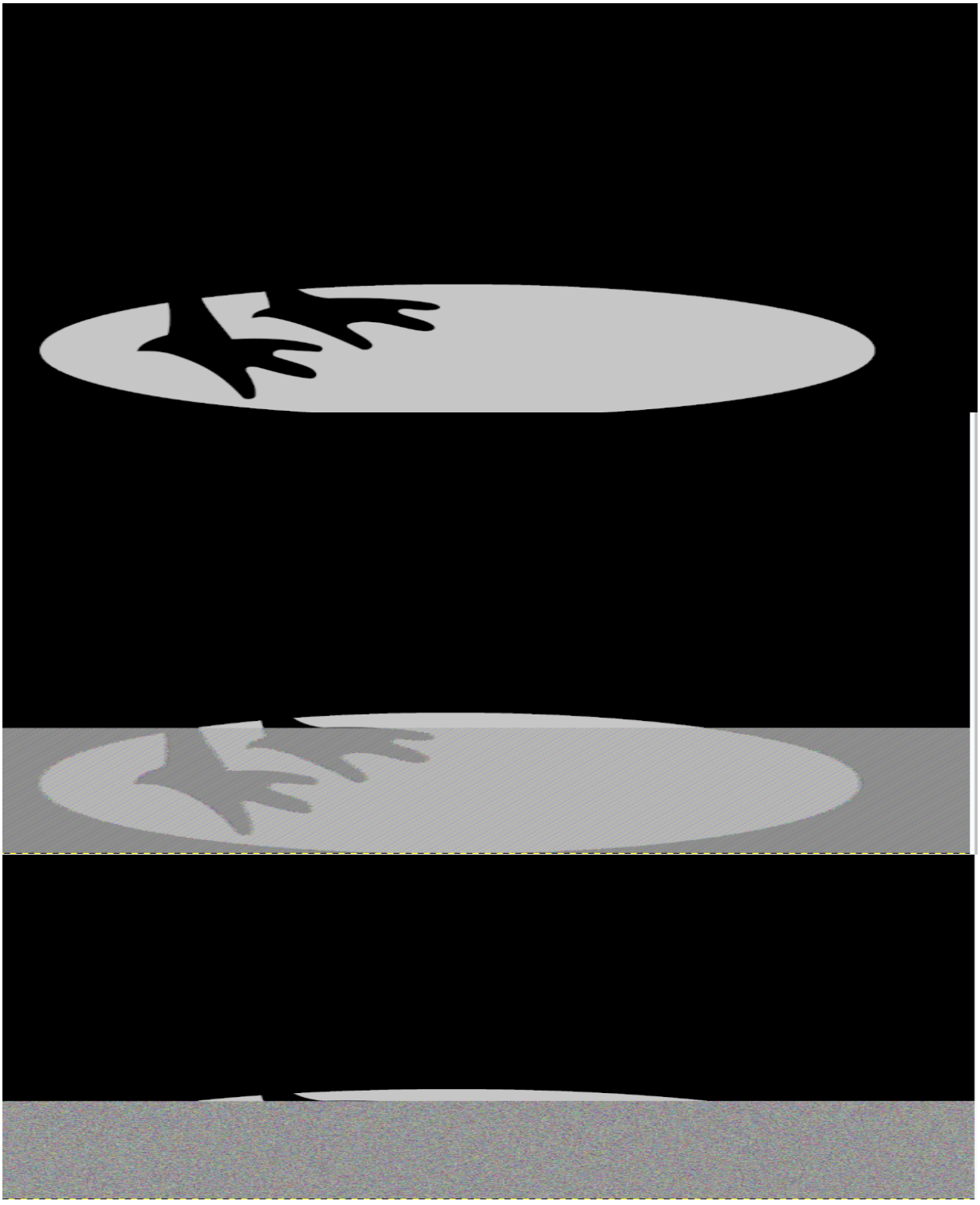
Çağatay Screenshots:




View Image Layer Colors Tools Filters Windows Help
filter
2. Hardness 050 (51 × 51)
Basic,
Spacing 10.0
Mode Normal
Opacity 100.0
Lock:
ubuntuc.ecb
View Image Layer Colors Tools Filters Windows Help
filter
2. Hardness 050 (51 × 51)
Basic,
Spacing 10.0
Mode Normal
Opacity 100.0
Lock:
ubuntuc.cbc

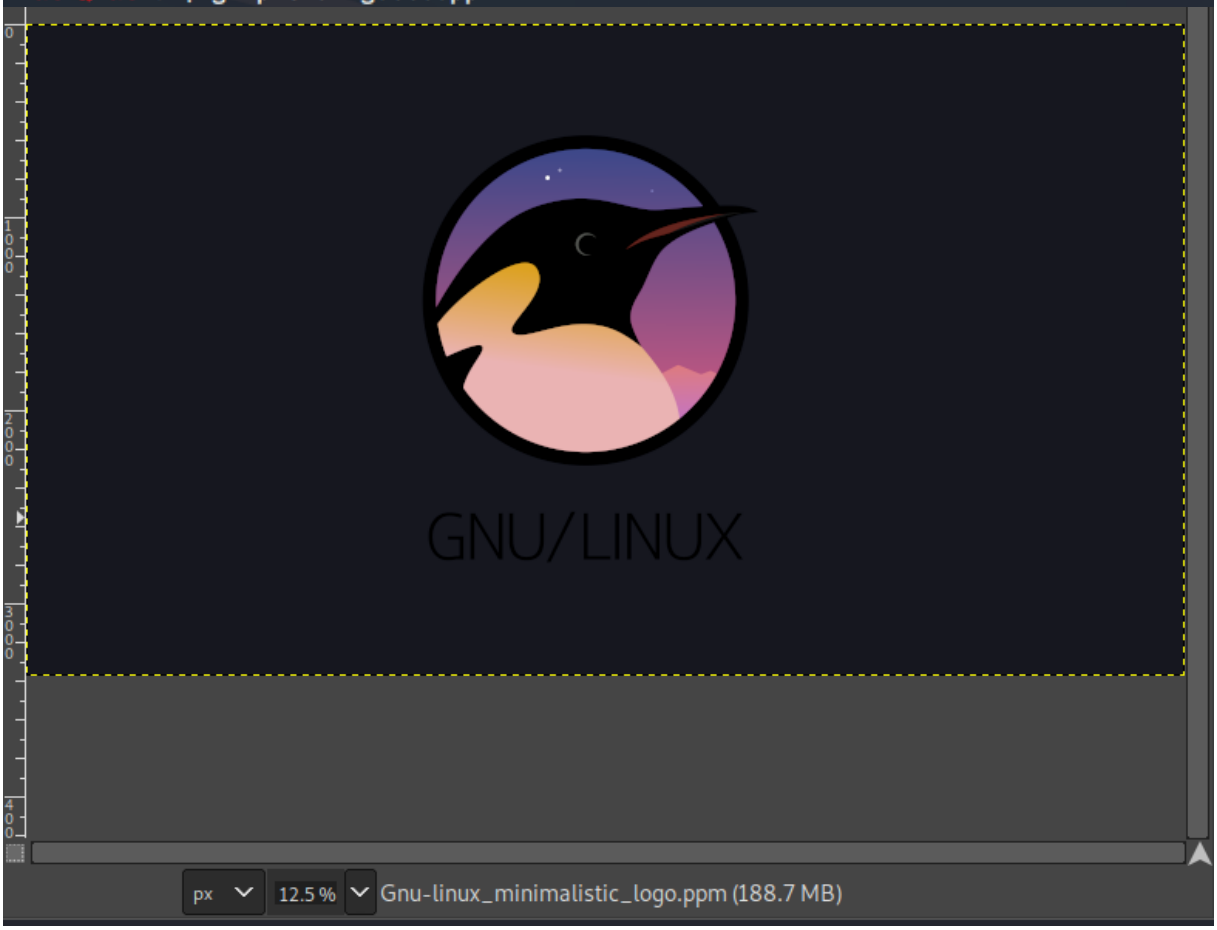
Berk ÖNDER Screenshots:

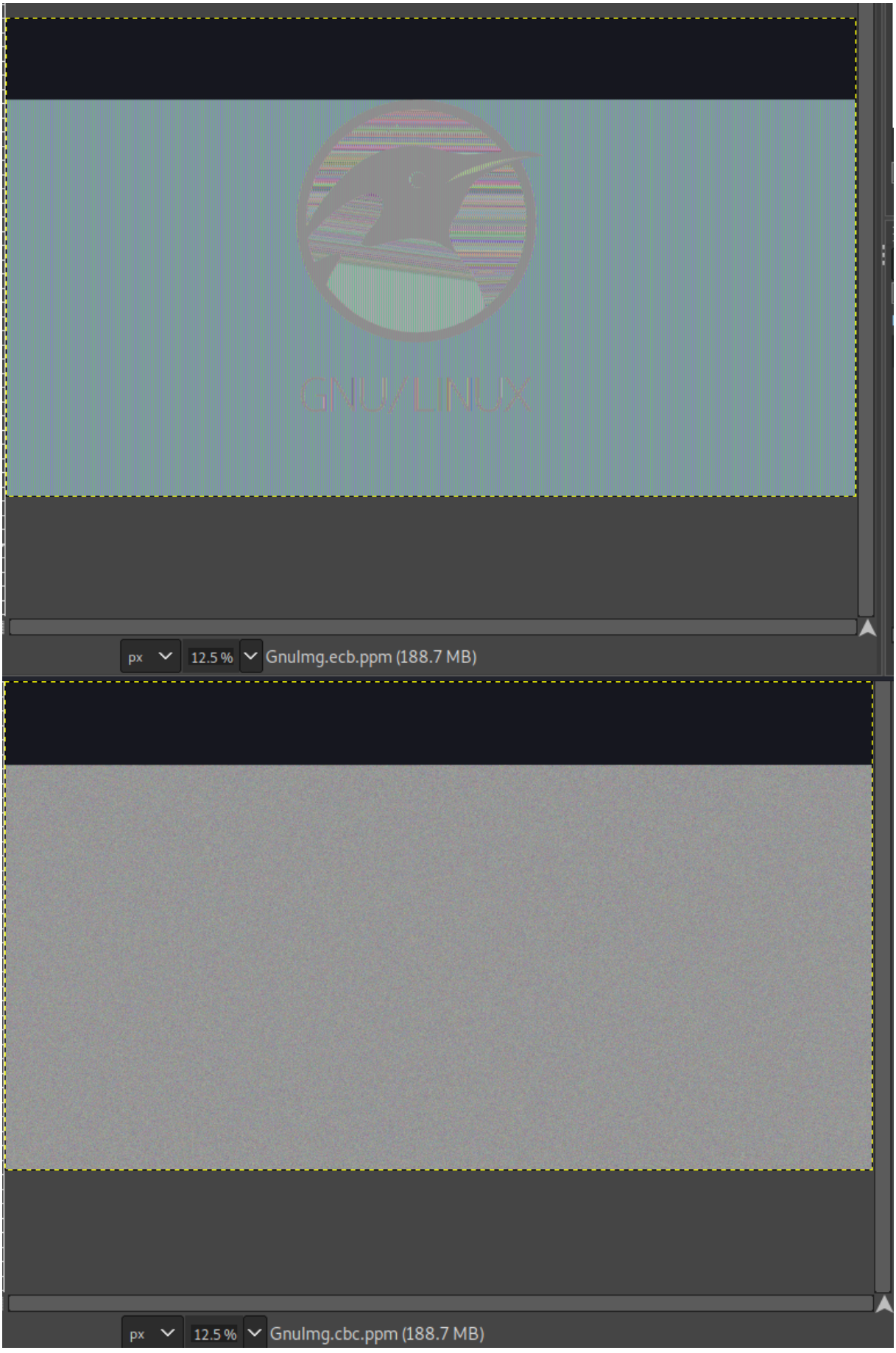




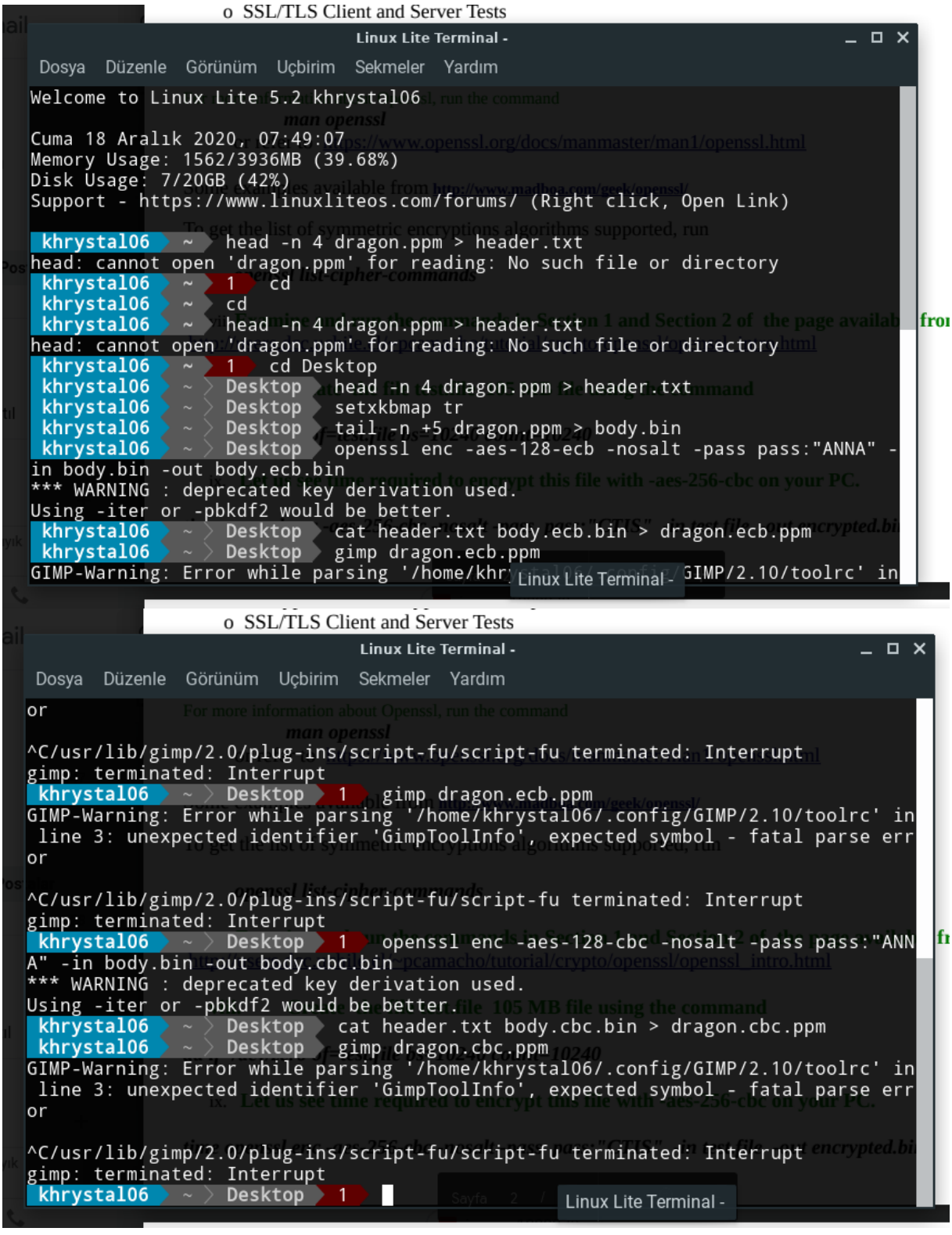
Bahadır Çetin Screenshots:

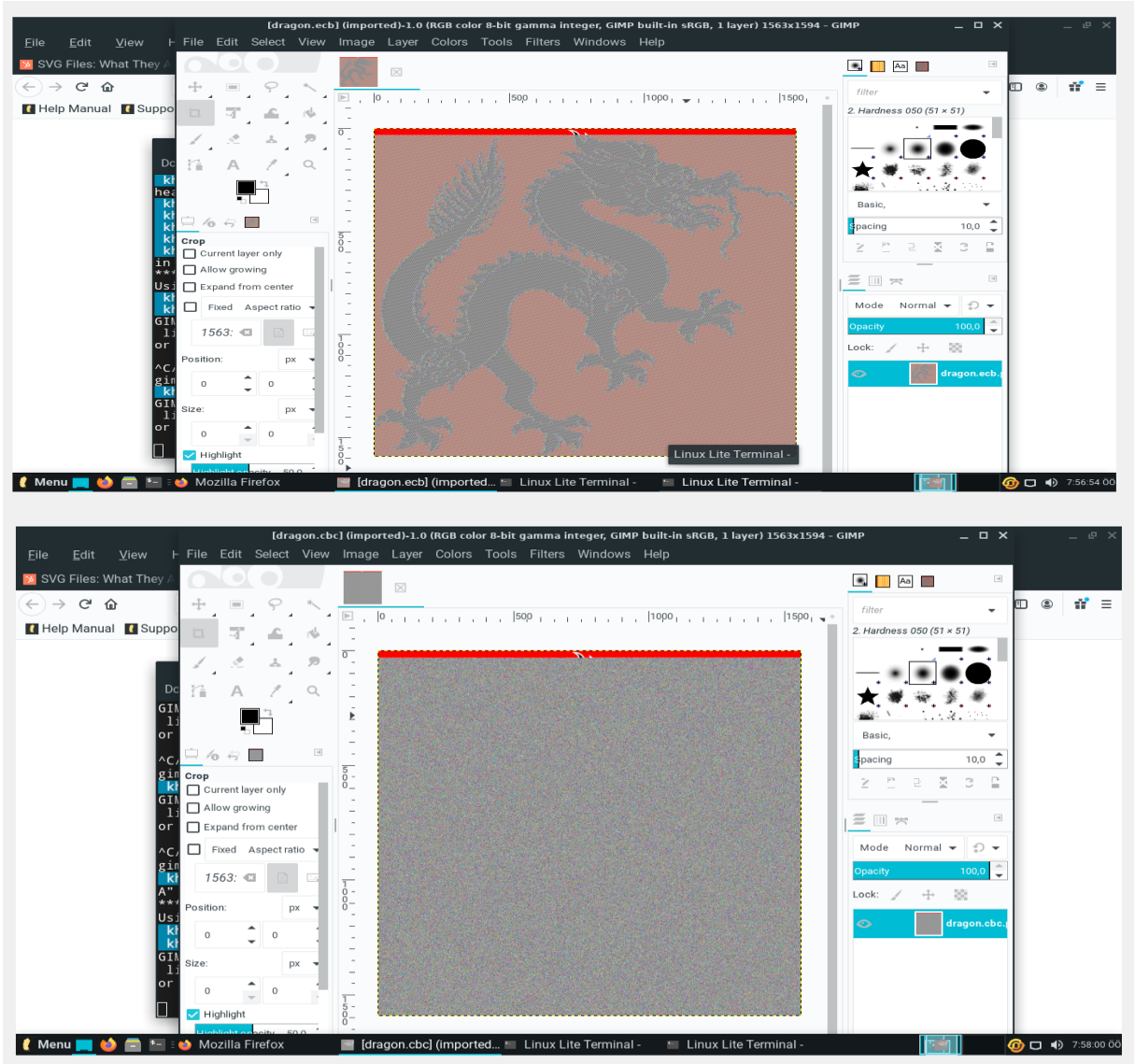
```
kali@kali:~$ openssl enc -aes-128-cbc -nosalt -pass pass:"ANNA" -in body.bi
n -out body.cbc.bin
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
kali@kali:~$ cat header.txt body.cbc.bin > GnuImg.cbc.ppm
kali@kali:~$ gimp GnuImg.cbc.ppm
kali@kali:~$ openssl enc -aes-128-cbc -nosalt -pass pass:"ANNA" -in body.bi
n -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.bi
n -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
```




GNU/LINUX
px
12.5 %
GnuImg.ecb.ppm (188.7 MB)
px
12.5 %
GnuImg.cbc.ppm (188.7 MB)

Armağan OĞUZ Screenshots:



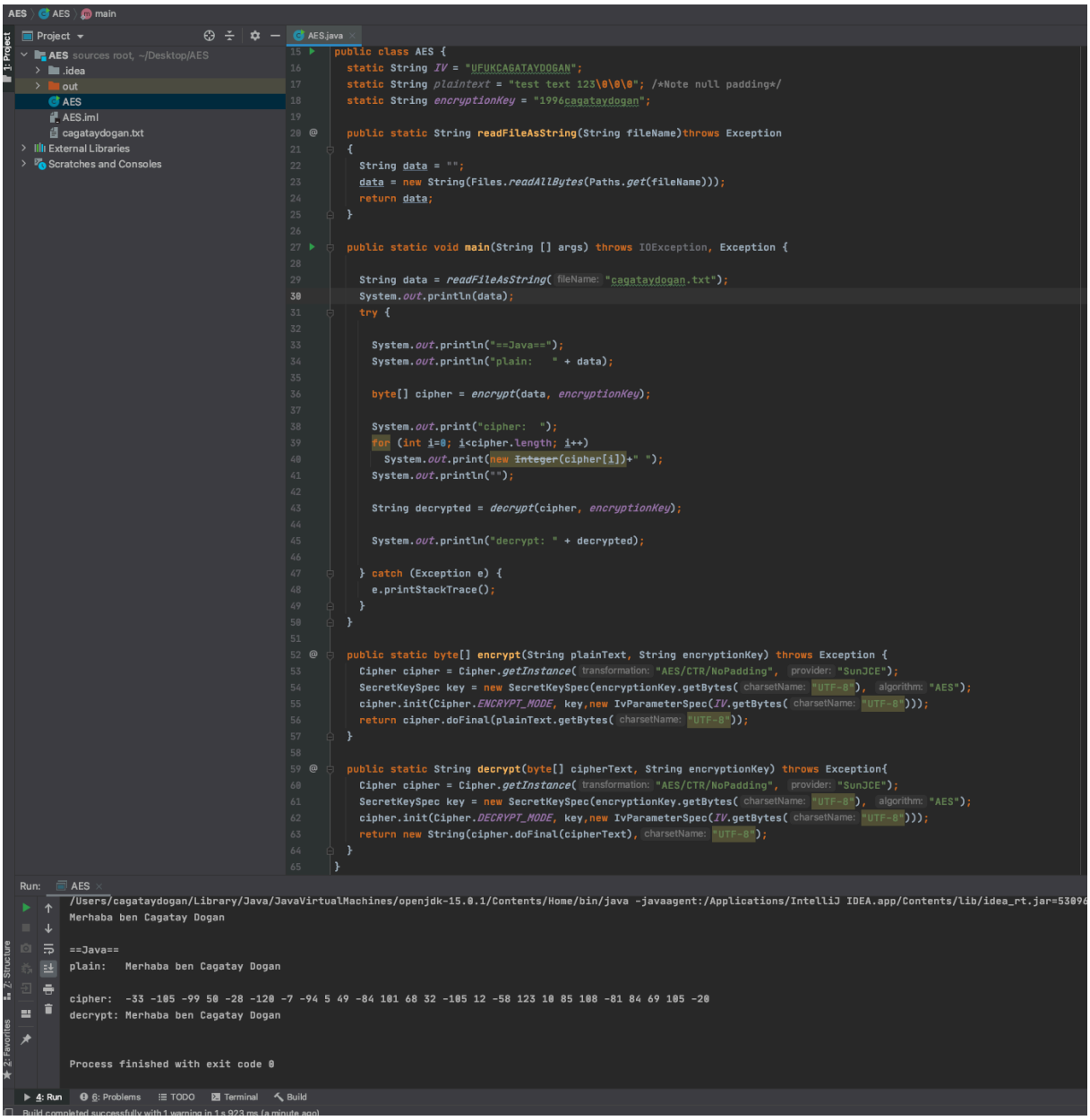

[dragon.ecb] (imported)-1.0 (RGB color 8-bit gamma integer, GIMP built-in sRGB, 1 layer) 1563x1594 - GIMP
File Edit Select View Image Layer Colors Tools Filters Windows Help
Crop
Current layer only
Allow growing
Expand from center
Fixed Aspect ratio
Position:
Size:
Highlight
Linux Lite Terminal -
Menu
Mozilla Firefox
[dragon.ecb] (imported...
Linux Lite Terminal -
Linux Lite Terminal -

[dragon.cbc] (imported)-1.0 (RGB color 8-bit gamma integer, GIMP built-in sRGB, 1 layer) 1563x1594 - GIMP
File Edit Select View Image Layer Colors Tools Filters Windows Help
Crop
Current layer only
Allow growing
Expand from center
Fixed Aspect ratio
Position:
Size:
Highlight
Menu
Mozilla Firefox
[dragon.cbc] (imported...
Linux Lite Terminal -
Linux Lite Terminal -

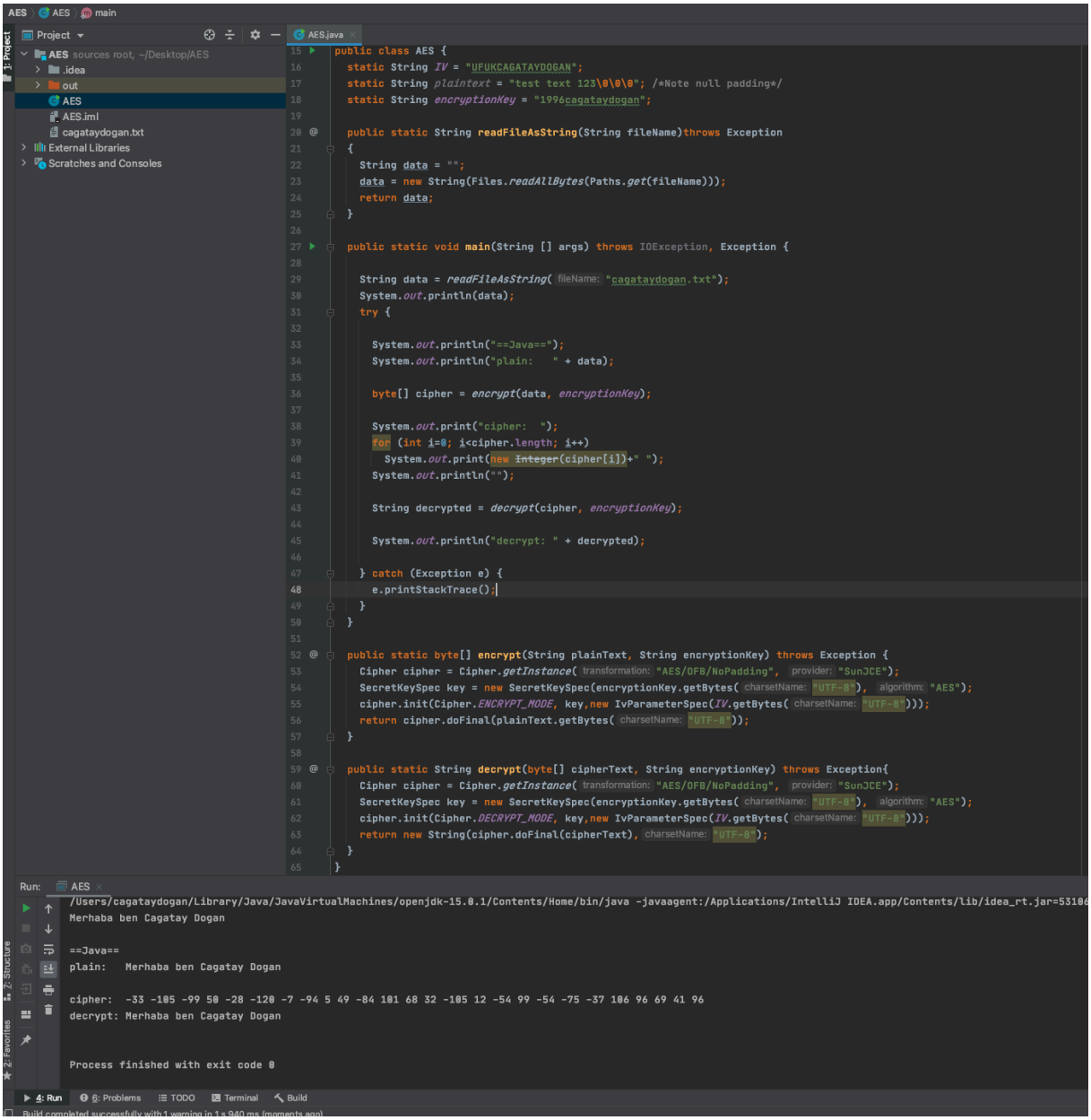
Modifying AES.java

Çağatay DOĞAN Screenshots:

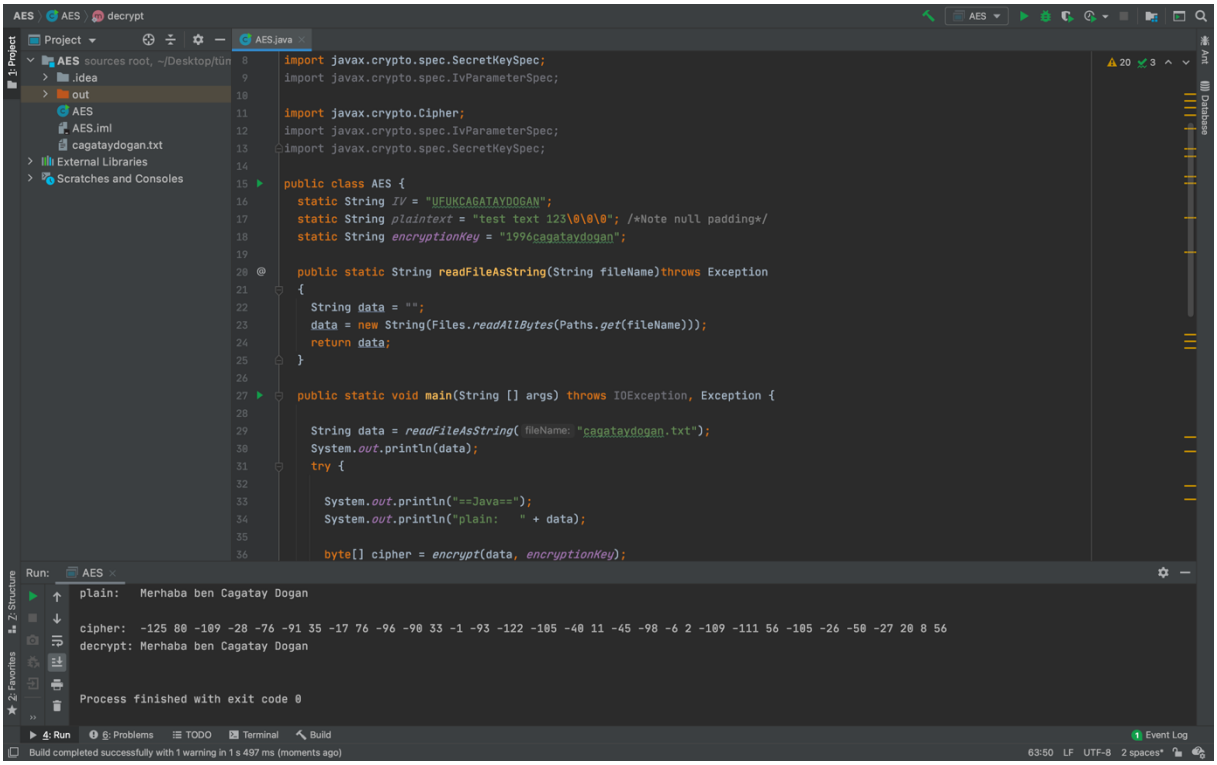
CTR

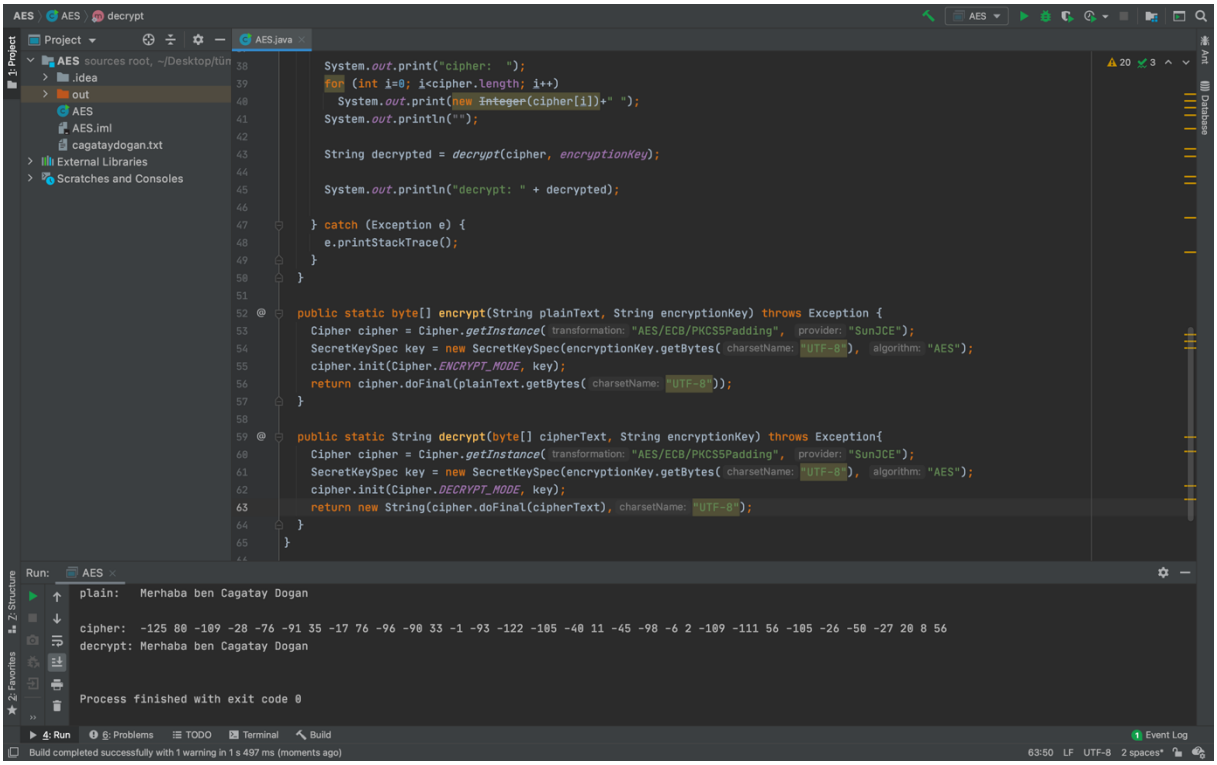


OFB


AES › AES › main
Project
AES sources root, ~/Desktop/AES
.idea
out
AES
AES.iml
cagataydogan.txt
External Libraries
Scratches and Consoles
AES.java
public class AES {
static String IV = "UFUKCAGATAYDOGAN";
static String plaintext = "test text 123\0\0\0"; /*Note null padding*/
static String encryptionKey = "1996cagataydogan";
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: "cagataydogan.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/cagataydogan/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
4: Run
6: Problems
TODO
Terminal
Build

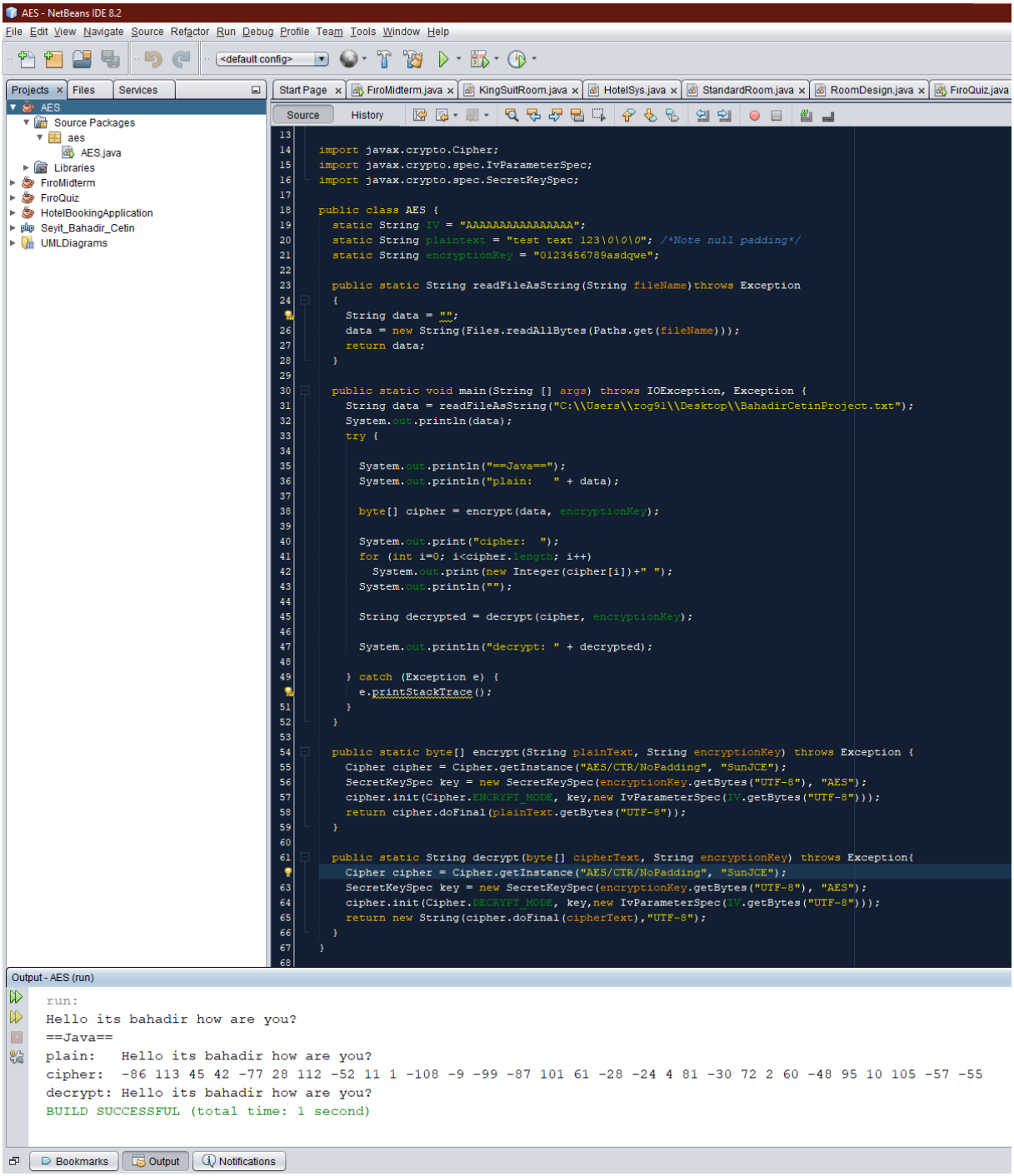
ECB


AES
AES
decrypt
Project
AES.java
AES sources root, ~/Desktop/tün
.idea
out
AES
AES.iml
cagataydogan.txt
External Libraries
Scratches and Consoles
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 = "UFUKCAGATAYDOGAN";
static String plaintext = "test text 123\0\0\0"; /*Note null padding*/
static String encryptionKey = "1996cagataydogan";
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: "cagataydogan.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
4: Run
6: Problems
TODO
Terminal
Build
Event Log
Build completed successfully with 1 warning in 1 s 497 ms (moments ago)
63:50
LF
UTF-8
2 spaces*

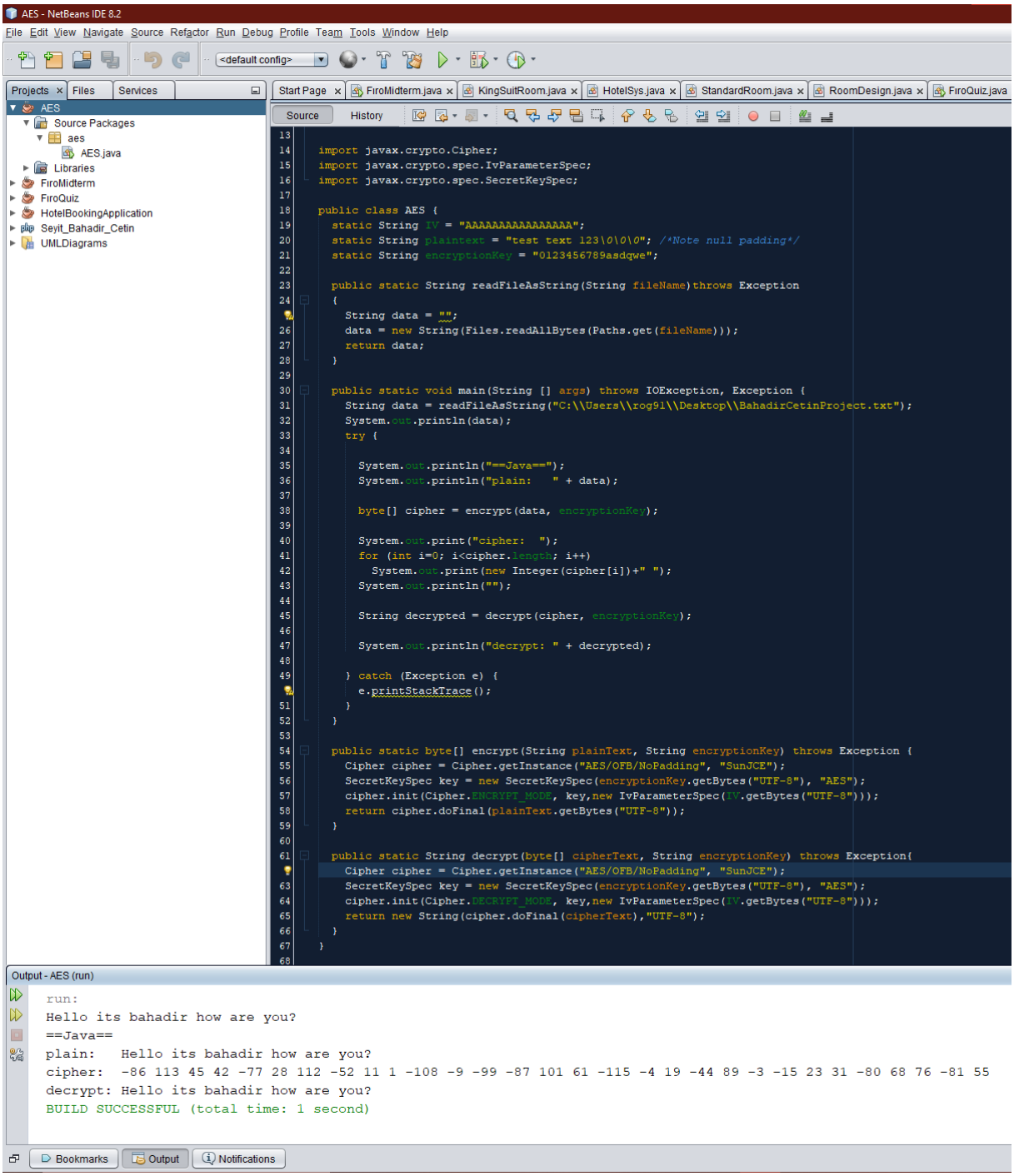

AES
AES
decrypt
Project
AES.java
AES sources root, ~/Desktop/tün
.idea
out
AES
AES.iml
cagataydogan.txt
External Libraries
Scratches and Consoles
System.out.print("cipher: ");
for (int i=0; i<cipher.length; i++)
System.out.print(new Integer(cipher[i])+" ");
System.out.println("");
String decrypted = 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");
cipher.init(Cipher.DECRYPT_MODE, key);
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
4: Run
6: Problems
TODO
Terminal
Build
Event Log
Build completed successfully with 1 warning in 1 s 497 ms (moments ago)
63:50
LF
UTF-8
2 spaces*

Bahadır ÇETİN Screenshots:

CTR



OFB


AES - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
Projects Files Services
AES
Source Packages
aes
AES.java
Libraries
FiroMidterm
FiroQuiz
HotelBookingApplication
Seyit_Bahadir_Cetin
UMLDiagrams
Start Page
FiroMidterm.java
KingSuitRoom.java
HotelSys.java
StandardRoom.java
RoomDesign.java
FiroQuiz.java
Source
History
import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
public class AES {
static String IV = "AAAAAAAAAAAAAAAA";
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/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")));
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: -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)
Bookmarks
Output
Notifications

ECB

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

public class AES {
  static String IV = "AAAAAAAAAAAAAAAA";
  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 {
      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
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
```

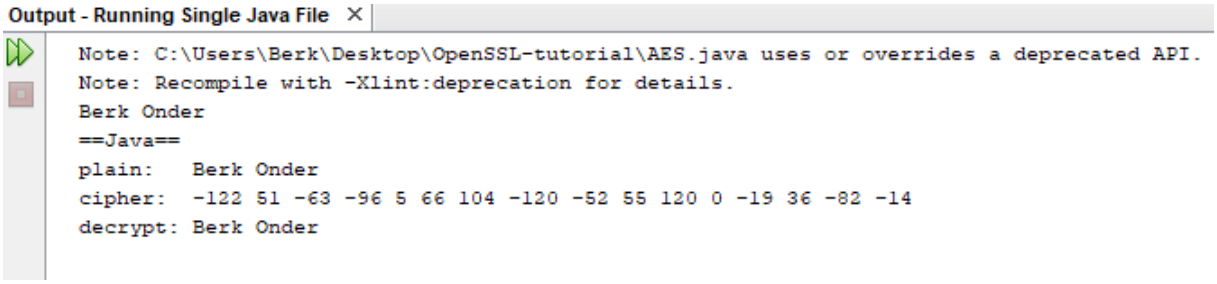
CODE

```
15  public class AES {
16    static String IV = "AAAAAAAAAAAAAAAA";
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    }
```

ECB

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

```
58    public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception{
60      Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding", "SunJCE");
61      SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"), "AES");
62      cipher.init(Cipher.DECRYPT_MODE, key);
63      return new String(cipher.doFinal(cipherText),"UTF-8");
64    }
  }
```


Output - Running Single Java File
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 = "AAAAAAAAAAAAAAAA";
  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")));
```

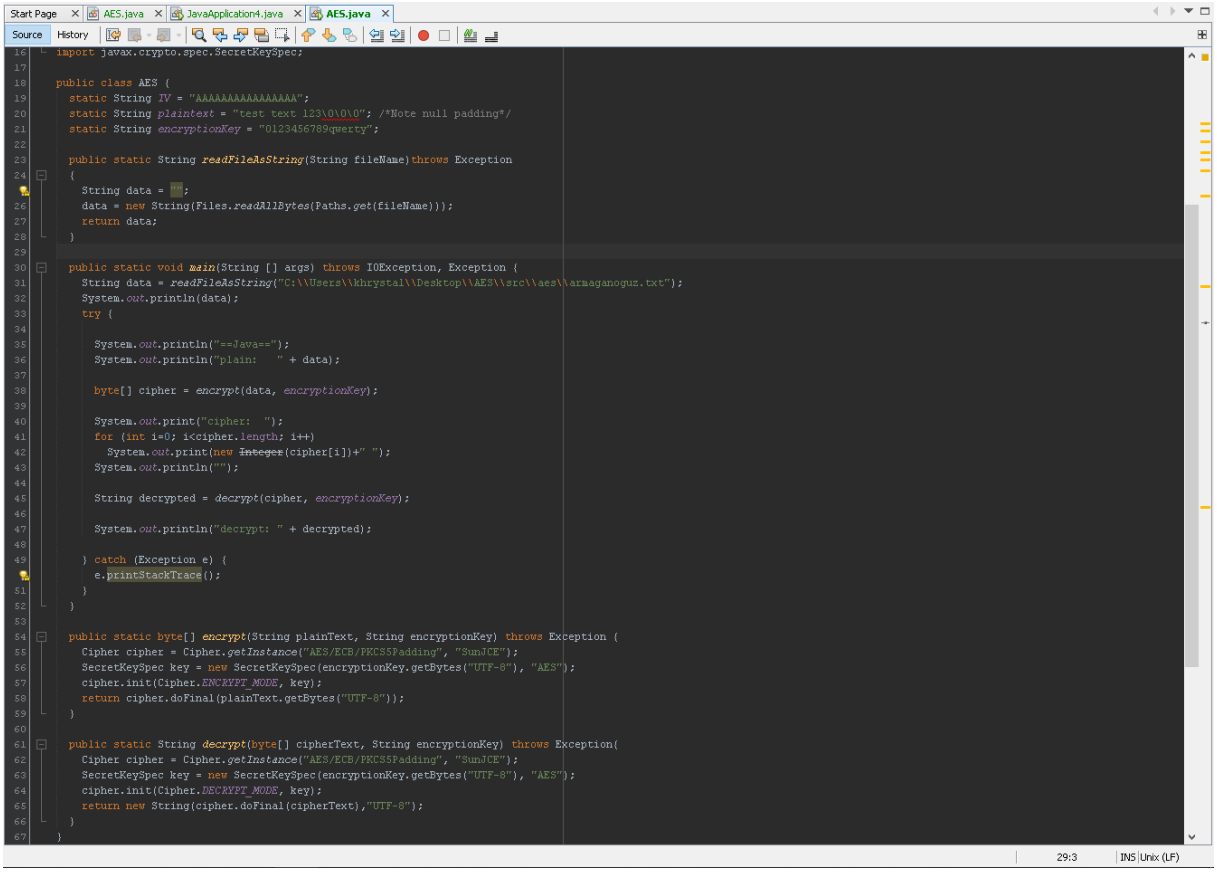
OFB

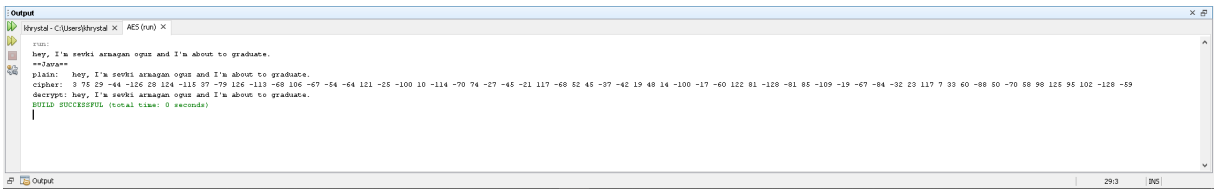
```
public class AES {
  static String IV = "AAAAAAAAAAAAAAAA";
  static String plaintext = "test text 123\0\0\0"; /*Note null padding*/
  static String encryptionKey = "0123456789abcdef";

  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/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


Start Page
AES.java
JavaApplication4.java
AES.java
Source
History
import javax.crypto.spec.SecretKeySpec;
public class AES {
static String IV = "AAAAAAAAAAAAAAAA";
static String plaintext = "test text 123\0\0\0"; /*Note null padding*/
static String encryptionKey = "0123456789qwerty";
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\\khrystal\\Desktop\\AES\\src\\aes\\armaganoguz.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");
}
}
29:3
INS
Unix (LF)


Output
khrystal - C:\Users\khrystal
AES (run)
run:
hey, I'm sevki armagan oguz and I'm about to graduate.
==Java==
plain: hey, I'm sevki armagan oguz and I'm about to graduate.
cipher: 3 75 29 -44 -126 28 124 -115 37 -79 126 -113 -68 106 -67 -54 -64 121 -25 -100 10 -114 -70 74 -27 -45 -21 117 -68 52 45 -37 -42 19 48 14 -100 -17 -60 122 81 -128 -81 85 -109 -19 -67 -84 -32 23 117 7 33 60 -88 50 -70 58 98 125 95 102 -128 -59
decrypt: hey, I'm sevki armagan oguz and I'm about to graduate.
BUILD SUCCESSFUL (total time: 0 seconds)
Output
29:3
INS

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.mozillazine.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

Bahadır 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