Wrap_socket metodu, *do_handshake_on_connect=False* parametresi belirtilmediği sürece handshake işlemini otomatik olarak başlatmaktadır. Bu nedenle kod içerisinde handshake'e özel herhangi bir kod gözükmemektedir. [1]

Kullanılan sertifikalar ve key'ler şu şekilde oluşturulmuştur [2]:

$openssl genrsa -aes256 -out ca.key 4096
$ openssl req -new -x509 -nodes -key ca.key -out ca.pem -days 1000

Create a key and a certificate signing request (CSR) on a machine:
$ openssl genrsa -out server.key 4096
$ openssl genrsa -out client.key 4096
$ openssl req -new -key server.key -out server.csr
$ openssl req -new -key client.key -out client.csr

Now you can sign the device key with your CA cert:
$ openssl x509 -CA ca.pem -CAkey ca.key -CAcreateserial -req -in server.csr -out server.pem -days 365
$ openssl x509 -CA ca.pem -CAkey ca.key -CAcreateserial -req -in client.csr -out client.pem -days 365

Kodun Çalıştırılması

Kullanılan SSL kütüphanesinin Windows'ta kurulum zorluğu nedeniyle projeye Linux'da devam ettik. Buna rağmen yeni kurulan bir Ubuntu sistemde kodun sorunsuz çalışması için şu paketlerin kurulumuna ihtiyaç vardı:

sudo apt-get install openssl
sudo apt-get install libssl-dev
sudo apt-get install libffi-dev
sudo apt-get install python3-openssl
sudo apt-get install python3-pip
sudo pip3 install pycrypto

Bu nedenle sizin de kodu kolaylıkla test edebilmeniz amacıyla 2 farklı sunucu açtık ve gerekli kurulumları yaparak projedeki tüm dosyaları yükledik. Uygun görürseniz test işlemini bu sunucular üzerinden yapabilirsiniz. (sunucu saatinin birkaç saat geri olmasından dolayı 'son değiştirme zamanı' şüphe uyandırabilir. Bu nedenle size

yolladığımız dosyaları sftp:// üzerinden bağlanıp tekrar yükleyebilirsiniz. Mevcut dosyalar root dizini altındadır.)

Sunucu IP'leri:
46.101.202.226
46.101.222.96

Kullanıcı: root
Şifre: 8Bq1610T

Koddaki bazı sabitler __init__ fonksiyonunda girilmiştir. Bu fonksiyonun içerisinden ilgili değerleri değiştirebilirsiniz. Örneğin gönderici de
self.filePath = ''
self.fileName = 'Data_Algorithms.pdf'

Kısımları değiştirilmelidir. Eğer gönderilecek dosya, kodun çalıştığı dizinde filePath boş olmalıdır. fileName ise gonderilmek istenen dosya adı ile değiştirilmelidir.

Alıcı tarafında ise:
self.loc = ‘’
Kısmı değiştirilebilir. Bu değişken, alınan dosyanın hangi dizine (location) kaydedileceğini belirtmektedir.

Hem gönderici hem de alicida ortak olan sertifika ve key parametreleri şu kısımdan değiştirilmelidir:
self.context.load_cert_chain(certfile='server.pem', keyfile='server.key')
self.context.load_verify_locations(cafile='ca.pem')

Ca.pem, her iki taraf için de aynıdır. Aşağıda belirtileceği üzere root ca'yı ifade etmektedir.

Açmış olduğumuz sunucularda test işlemi ayrı olarak bob.key, bob.pem ve bobca.pem dosyaları bulunmaktadır. Hangi durumlarda hata durumlarının ortaya çıkacağını görmek amacıyla bunlar kullanılabilir.

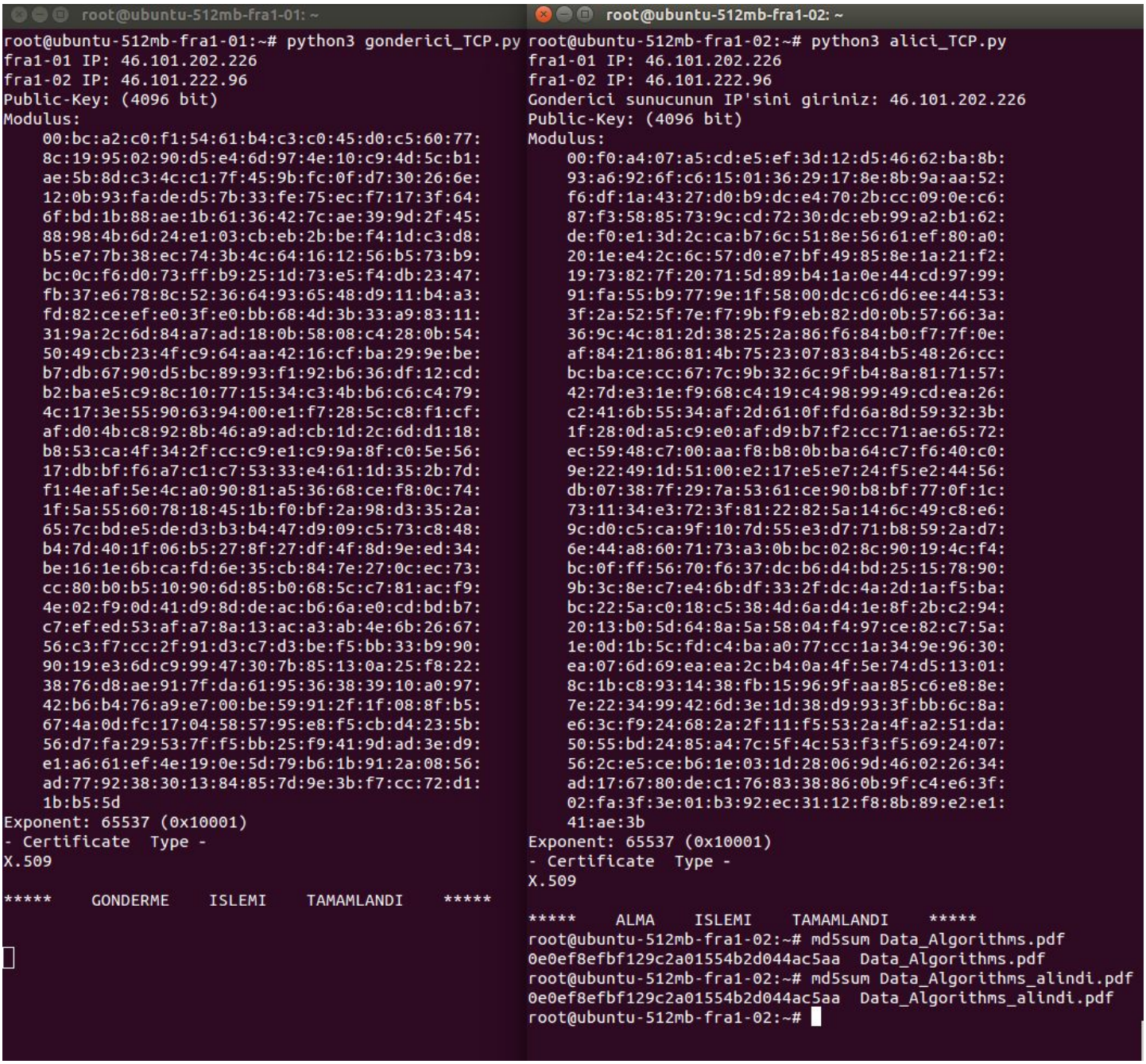
Kod test edilirken önce gönderici taraf çalıştırılmadır. Ardından alici taraf çalıştırılıp ip bilgisi girilerek test edilebilir.

Dosyanın okunan son parçası gönderildikten sonra, (veri okunamadığından dolayı) en son karşı tarafa boş bir veri gönderiliyor. Bu durum alici için sorun olmuyor çünkü gelen veri boş olmadığı sürece dosyaya yazılıyor. Ancak ssl kütüphenesindeki implementasyon nedeniyle karşı tarafa boş bir veri (“ “) gönderilemiyor.

Böyle bir durumla karşılaşıldığında EOF exception veriliyor (ssl.SSLEOFError: EOF occurred in violation of protocol). Bu durumun önüne geçmek amacıyla ikinci projede, aktarımın tamamlandığında kullandığımız gibi, gönderilecek veriyi “ “ olarak değil de “Done” olarak gönderdik. Böylece bu hatanın önüne geçilmiş olduk.

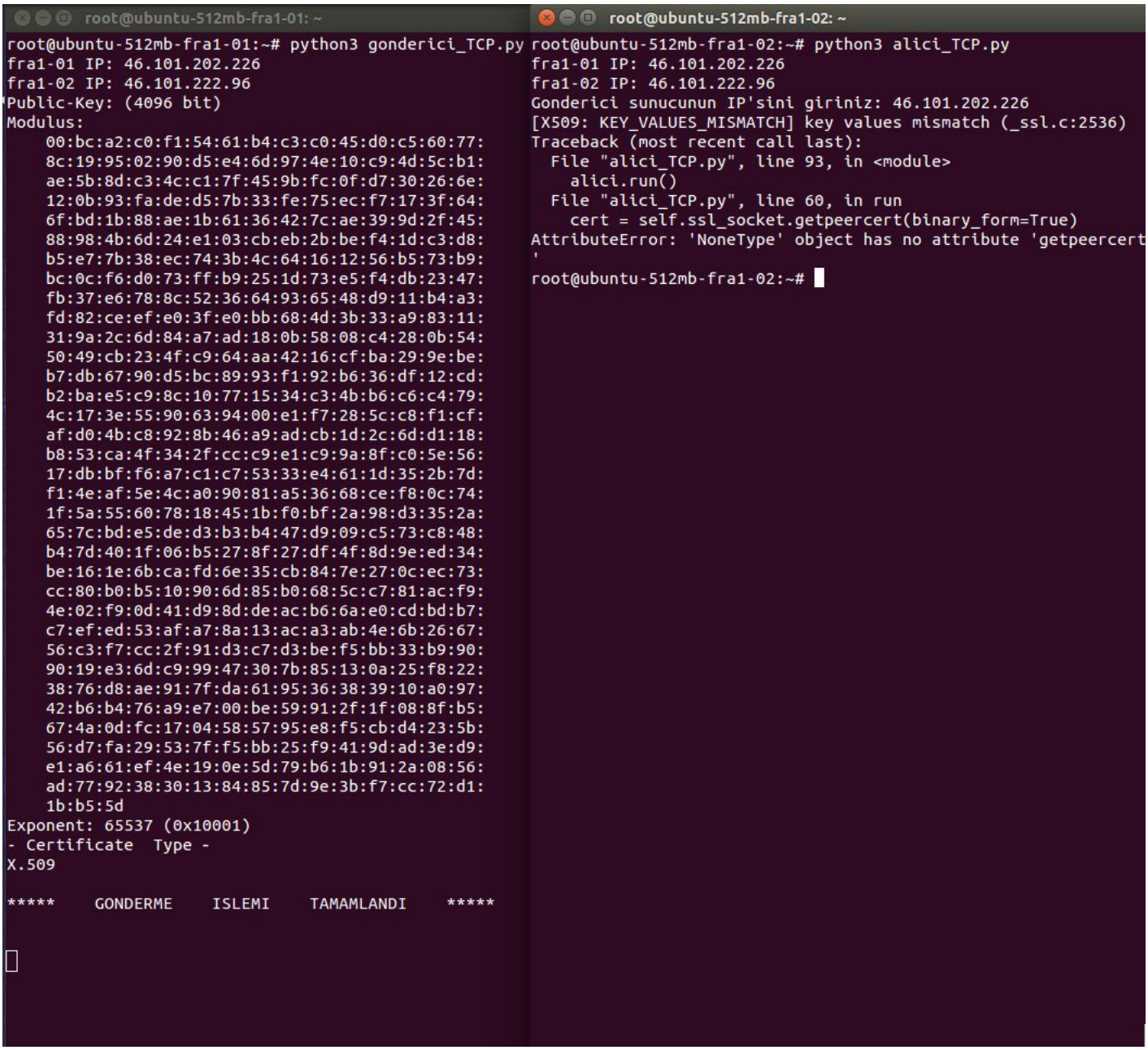
İlgili resimler aşağıdadır..

Resim-1: Başarlı bir dosya transferi örneği



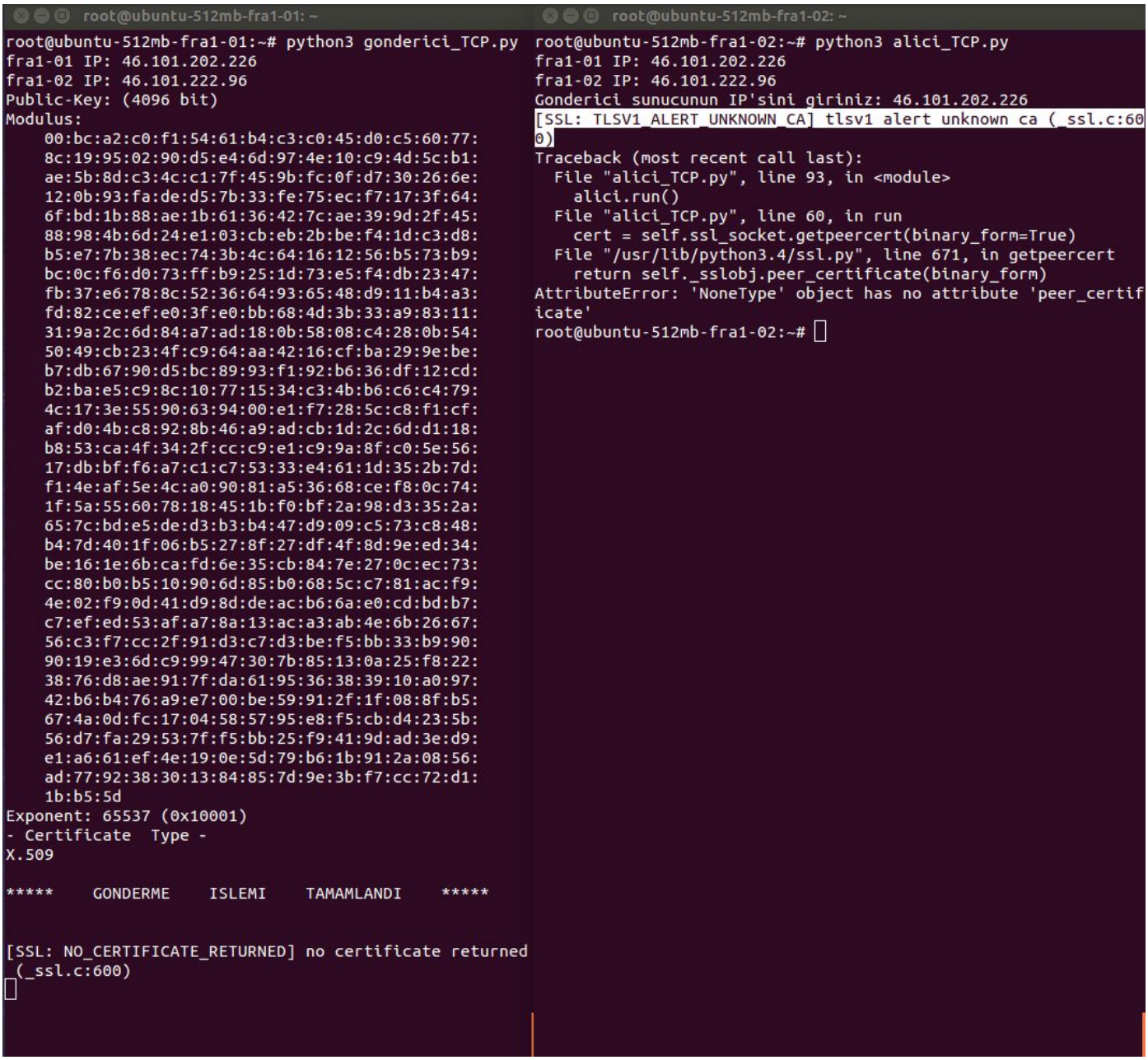
Resim-2: Alicidaki keyfile, bob.key ile değiştirilmiştir.
[X509: KEY_VALUES_MISMATCH] key values mismatch

Değiştirilen key, sertifika ile uyuşmadığın bu istisna meydana gelmiştir.



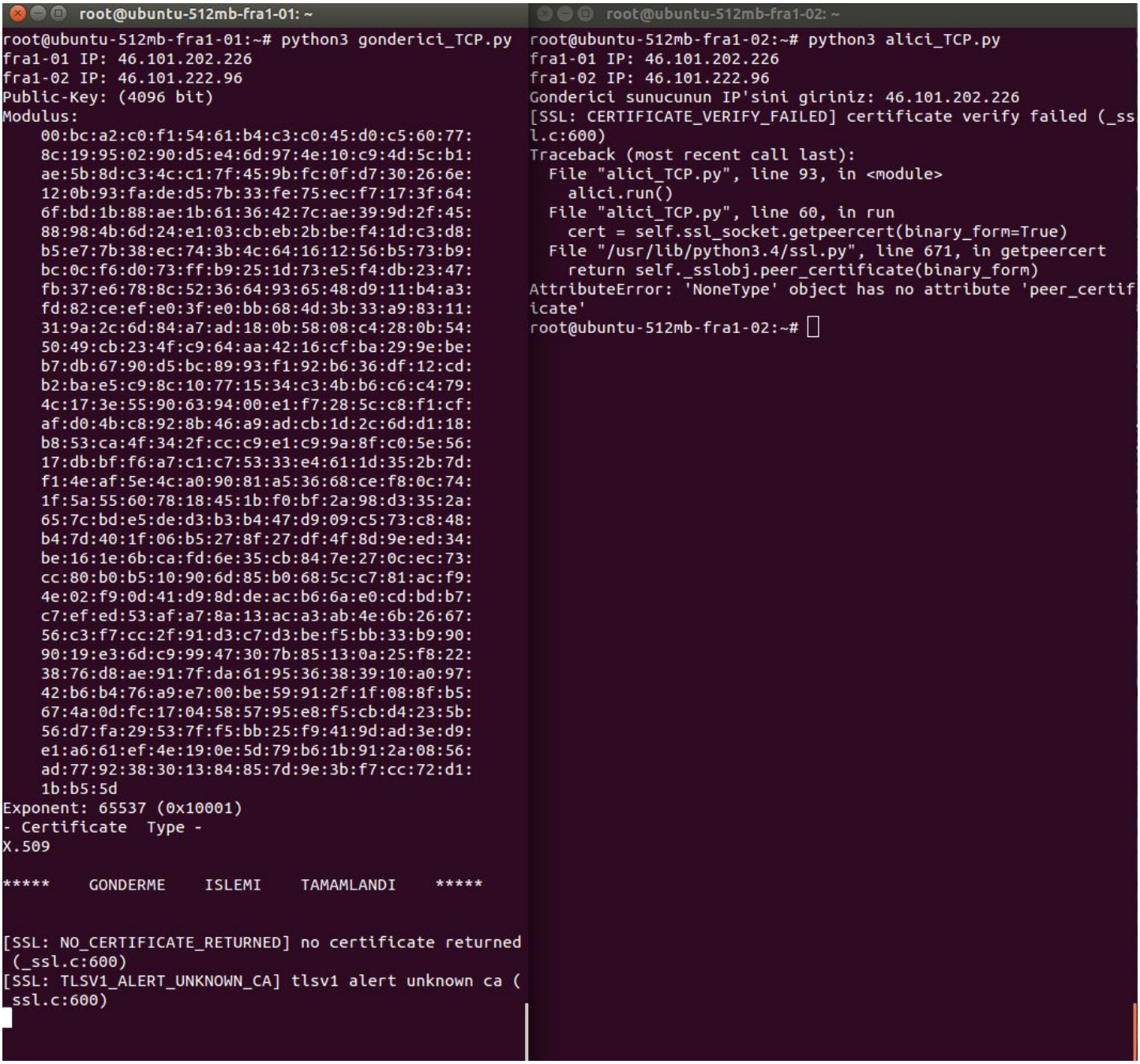
Resim-3: Alicida hem keyfile bob.key ile, certfile da bob.pem ile değiştirilmiştir. [SSL: TLSV1_ALERT_UNKNOWN_CA] tlsv1 alert unknown ca (_ssl.c:600)

Artık key ve sertifika uyumludur ancak durum ca file ile çelişmektedir. Bu nedenle bu istisna meydana gelmiştir.

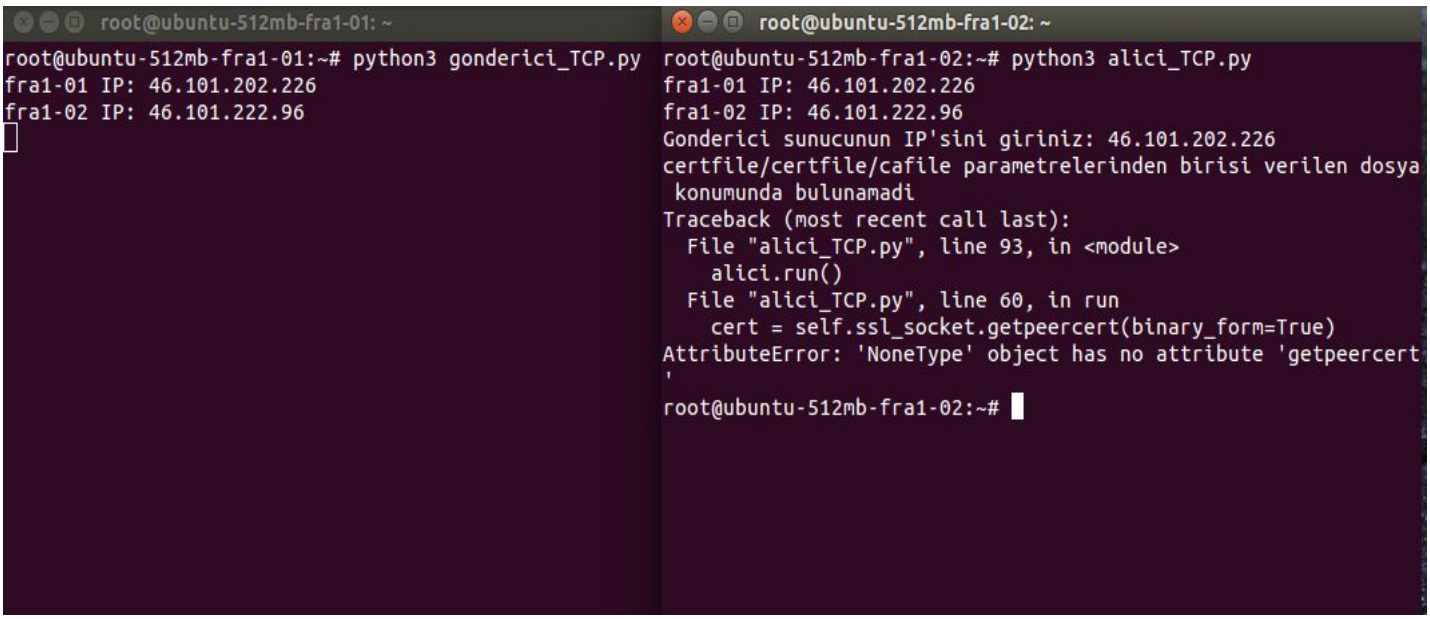


Resim-4: Alicida hem keyfile bob.key ile, certfile bob.pem ve cafile da bobca.pem ile değiştirilmiştir.
[SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed (_ssl.c:600)
Gönderici de ‘tlsv1 alert unknown ca’ exception’ı gözükmüştür.

Artık alici tarafındaki her şey birbiriyle uyumludur ancak gonderici tarafında tanınmadığı için bu istisna meydana gelmiştir.



Resim-5: Alicidaki keyfile, mevcut olmayan bir dosya olarak verilmiştir
[certfile/certfile/cafile parametrelerinden birisi verilen dosya konumunda bulunamadi]



1)https://docs.python.org/3/library/ssl.html#ssl.SSLContext.wrap_socket
2)http://crypto.stackexchange.com/questions/26591/tls-encryption-with-a-self-signed-pki-and-python-s-asyncio-module