CS6903: Network Security

Openssl Tutorial

PLAGIARISM STATEMENT

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all materials and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarised the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honor violations by other students if I become aware of them.

Name: Abburi Venkata Sai Mahesh, Khalid Shareef

Date: 06/02/2021

Signature: CS18BTECH11001, CS18BTECH11029

Description:

CS18BTECH11001 (Mahesh) has played the role of Alice and Charlie and CS18BTECH11029 (Khalid) has played the role of Bob. To Understand the process much clearly we have depicted it as an interaction between the Players (Alice, Bob, Charlie) in a tabulated format for each step.

PART A

Alice	Bob

Creating private key and encrypting with a password for Alice
 	
$ openssl genpkey -aes256 -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out Alice01-private-key.pem
.....+++++
.....+++++
// Enter PEM pass phrase:edith@4
// Verifying - Enter PEM pass phrase:edith@4
	Creating private key(also contains public key) and encrypting with a password for Bob
	
	$ openssl genpkey -aes256 -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out Bob29-private-key.pem
	.....+++++
	.....+++++
	// Enter PEM pass phrase:khalid
	// Verifying - Enter PEM pass phrase:khalid

Viewing Encrypted private key of Alice $ cat Alice01-private-key.pem -----BEGIN ENCRYPTED PRIVATE KEY----- MIIFLTBXBgkqhkiG9w0BBQ0wSjApBgkqhkiG9w0BBQwwHAQIu+WDfnssOG4CAggA MAwGCCqGSIb3DQIJBQAwHQYJYIZIAWUDBAEqBBCz6cDnfbrtSsSUffMkpqQoBIIE 0F1qgYFv6ufodUZuuzU/MW9FDt8uvzY6hyd3pfbc5/sDr57/6A7x9XnaATn55SfG 5T82mpAHNanXiDUtE+W3QC0bPFgCmT/V8CchCfWFbGW2dRSuNo/OXh+yvMitmL4J +esl9pKVnJ2eDi1MVPIGn0V9ok1Qn5fB+TXOfVQ/r4zjFGfwwxLulFEounjT0aIS o5zMNZTdVvwoo8hUAyfU1yJoett+X1mEa4jMFWcwsbcV+Hl4iaCzvLEDCk+YAD0B bGCzL9v4M3rMycPcUP3JQ2WTrIL5sDCg/Kts2jRfln6OYqhDTAGKaLt/UDm4ulf2 RD4tT+HCELqW2tYPVMvVxXSZMwUyIXVcTVTLUKFLm1abbXjAxZV9Bt0g5Qgw0wE1 3W7i2mXhU9YEpbIi3zy0/G2BbIbEH67gcA3fI5D4Qy1Xbr2LmfRWwctGEM/SJtM4 gUopFU6vGjSqUh15jvedb2L2int55OxY44mYAYqce6O1JzJN5ovXwEcqC4w+QEEA exT6HL3CSybGt18INI91g73UjrYVacw4PhOat/pIAKQ8iC/AJlUoYWCk6qkEYt/e qJ2p0eWm3j4ql02n6evhckgAb67cwzsZJTYBZJJ9Ld9t9RJwkJYDAs7qqYjPg0Iw C5sU91Tu1ibHEHBeewSpLvVbLYJRGci0Lt8iRwv3SZA74j1c783jJRxLB5udFgES KY2CMoOAnFzNxQaO5AIZUxmAbAbFvPyluakaDqDjiabIXwosCQjKGPEjdjYrip7G PuqVuv8rA+zkDThTCdDF0Hpkgcge5uhS4COBXp9g2	Viewing Encrypted private key of Bob $ cat Bob29-private-key.pem -----BEGIN ENCRYPTED PRIVATE KEY----- MIIFLTBXBgkqhkiG9w0BBQ0wSjApBgkqhkiG9w0BBQwwHAQI2ZejTFyYcbMCAggA MAwGCCqGSIb3DQIJBQAwHQYJYIZIAWUDBAEqBBDvS40RyZz9Mox3V8FhbiYPBIIE 0B1LgvYYa02+wXaMwzp2bSTJ/eERXuBGMhpQ1ZrrGPHwAi8yY38S93eHU5alUp0a R/aks+nJNyzjI9lyTfZTqitUMQ05i1H5Baealk/4ydtwFjaWrlgK+cT2UXO7lpS4 ZtV3p7SI/rarxQem5Lq3GtA4xzyKxy1N9Wppc4bj0bJBV5fGfinFX54m3KUQgHkU UUiKsRC4TS80afBw5Q5njFTJMYr7AlTLt/EQ9BiFZuAgLxUcJ2LmDsb/HOA2Q2xU CRGj4dOCVE7QB/5nfuuUSgxuIF7gs5Zj3JgKShDGYZzHLEXlFdUUSG8XMLopka2i QA/Qb0VOU6X40Oato5vAIx9mo8E+dYEpKrLi753bidX50So8omCgHte2JAdToovQ Ai9mpuEHH8b7/ZaUd3PJj+ECw7RymPwOpEAobjjQAWzoslQfmQbz6eDsE/ra7JuL t+jk9cFqarTO4B+jgf4CwmcfMoDedzAR83ltFYNzI4+wot73yEEeFsinV2HfdfPs YpPiIJlOrwo5q0JPYZ6CTGzodbxN7Sjv9b5W219L1Qq5b6AH4Irqs8me/10G0qqO r0HbduEhT60/eg1rmXH6gdATMrlUh9ooPGbzU4TmkHvl1U2HPa8jExenN6hSDfQ6 wfgk0xfX4lHPeWSI7zB/7Hsw7mqDnEJlOzAgYb9MhoU+yH+Ua2Vi6W53JWfsHbiD xkD6d+UT5TfDudJY9FHDObS7K4ehY6k6/LpNTV9gSLrk7hL8wNY6n3ULhdrWkxEP OJ/vK3agGAIodv88MZQdGMa0wBdkqSjvzljGtfbysHOsR

cc1vqQEoaAZO6GsVYihO97Z An0ZjJdUTkgGFEHe9mRNxcCD02Gb/R+8h+ULNf+FLc KSw/g1SBFSNwJoUTEv6Faj lpz8DIFIlZQVtl4h7HAYZdk+h2P3zRgyQll3SewzOT5qU 4Jh6gd/iN4trt9kha4s CZnOGi91lqH/c9OfCtSX2UwVnv9tZ78/SpyijAVUN41elI gpJ/AR3iQa2VtULOm+ 9qX6Z/3S7E6DU6FPPkTElMQPNFHnbYG3/54b86Bl/k Z56pkYU38f2MXaU8nhlhZ5 LEHrcXu0mx23hf+7rDEvcaOyBSliny98VuHC/fKaX7Q3 L36AKcOuXjlQPdqQ0eAl wE1En2pBygKlEKMRCtaaunWid3qalLV7EBig3990Am9 9rtPqxW1yozmC7a4jQC8y 02IXQeKvhSd+KBDwelXBZlr+/lbXXpEbnXvXEphPMHP RfRZDQu2sBk3RX75Anld7 72CgvocuPMaNqAKp5UbRG1BF86HBNhyVRcFUhDSk yUzQtq4eo18Lu8hHPXml65uE kNJ0wTL/l+fDrEmlGelBjCxYRtZK5SbCfvtXuUMH4PEtV g3ct+OPSypJytXNB+lr r56/k6s1VW1x2Y96ed6Jr5s+DDZkeDvKZfkbo41K0a8y CtSaFyOlqLkio7DZqlJ3 PdCSa/dZILGyball1OH28XJ2Me05AxGezhpO1nBgCyw cvCKI+KG61wcYmvdbGFFa DN7YffYIoV/jKJQtcmOxjcuOUYkrRJYdQjDnHflkpb2Sx vibJsaflxBCjd1HUTF4 u9pPehy9GCDqNIS0bpQOMOkxyvdtzuR+hCDp3cMF Xghb -----END ENCRYPTED PRIVATE KEY-----	tC7PSBSI6i9q6/8Z3oQ pQrWL69qeQfpmzAdauMRUw1istQhKzerLRLA6b/Y7kXb Zligv7wOnsFB2rnsycIM 7OuqrwXuq6OL8zM5TpD0iJDMs8gactbDr2UYT1MxfQJ e/18KBSTEHXxczhBed7Q6 E8eXg32XM2zg/qh1dlo502L3WP1CSCSwZL+2bTiRsukv 2p/TBoPkwXeLlQ6XbyPB qbCcoq6oNilhJCaOJgCYjgSZFGq03mbC/1hUvSxALxxFL 5m9PeScaAkC5VwLhux6 OxxeLYp6Km2VClu/hb3E+fj8XRyaoMw9C5UKUEQ78uL HrQFm+ikNqpXJIBtMPtQd cXJFuWvF+KLiyViQZj1lcFq6UnXuUnWBGJgyKpLxMn3D A3nc8DH9ra2oeXuHOOJH RgXfal7ZjGkSCS6SibBeu2qktzU1BH2hpMYxWjC3GlqM WMWUQO7OAv34fSPtf0lc XIMKinlyKAlVI0C3KLXyxVTNJphA1fks0SZsbCgTPydQHZ La/h/t8yj4wVtqr8tB tCoB9rSQDgYbEVW8qfxrUqdeprff7jAMHxCtvdwxyjplgf1 2282pm3HlKxyof1dJ fhw1ElMYUnoiUrwDma9VPRFH9+pi8zfRMzbocsGLQV41 oDgUZ7jZSZMPAZYwoVoH qK5ws9vzjqkJgoM/+OvG/YkbLIPNI+EggoEA79KWFr2DX bdxCMZvkuyLsA4cfeh6 TNUBWFJAJWSXs5E9DzW7/K9BQrcn2/GbwxIel1DrxcDh 5qYjVtQd7pRPOxRseYY4 qpZ0dBl0M7XJZq8YFY7B7Ayu9kpXQ1wVZmsQmAiyFliL -----END ENCRYPTED PRIVATE KEY-----
Generating corresponding public key for Alice $ openssl pkey -in Alice01-private-key.pem -out Alice01-public-key.pem -pubout \\ Enter pass phrase for Alice01-private-key.pem:edith@4	ThGenerating corresponding public key for Bob $ openssl pkey -in Bob29-private-key.pem -out Bob29-public-key.pem -pubout \\ Enter pass phrase for Bob29-private-key.pem:khalid
Viewing Public key of Alice $ openssl pkey -in Alice01-public-key.pem -pubin -text -----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCA QEAltsvZ+4eii5C/6PuvuED V8UX8muY1fm5CuTe1qg6lnQyBB6kz84EJs9wYq30Vq yiY0uf6Mmhg1R8M/MaCOYs zCRUNwcJW4F4ozNJKL5reJNDWTDMmSchyvMhsPEe X9hLyWi1z4b5nV1KNPATHyfB yNv1eoqaiRzB6EmFcwjn5XWFwov6sTLGw4Xs5YgPLT	Viewing Public key of Bob $ openssl pkey -in Bob29-public-key.pem -pubin -text -----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQ EAwwhE4SvrZOrVMWVccQbM tHbhqI6n2dLGNpz9guPSttpHOqbhjIrWFRGh+wu8rzfsOo aUTOSOxHRQIIPckrmP Iiba9+ybuBYbowVtRBY65IF/iGMbjj7cvkAvgOm2NVeLAB GuMpJEcDBfv6QzwRG2 6Y5tD7M14PxXhtEDpnNjL/mviitR+LxLGhsHp79lNV27ziaJ

k9TpaKR9B9DsiG8TF93pp1 fzjvBlrtMqSvvWSLwA05PgaoT/aLwAQ39npTaOu56pIJ iQxjWgIrJ0K7ghi5mM2Y EYuudDBJ46kBtHlAwJDEstx+8/dXC16lCdS44qEbkI+fm dsZs0IqEECO85AH/cxN EQIDAQAB -----END PUBLIC KEY----- RSA Public-Key: (2048 bit) Modulus: 00:96:db:2f:67:ee:1e:8a:2e:42:ff:a3:ee:be:e1: 03:57:c5:17:f2:6b:98:d5:f9:b9:0a:e4:de:d6:a8: 3a:96:74:32:04:1e:a4:cf:ce:04:26:cf:70:62:ad: f4:56:ac:a2:63:4b:9f:e8:c9:a1:83:54:7c:33:f3: 1a:08:e6:2c:cc:24:54:37:07:09:5b:81:78:a3:33: 49:28:be:6b:78:93:43:59:30:cc:99:27:21:ca:f3: 21:b0:f1:1e:5f:d8:4b:c9:68:b5:cf:86:f9:9d:5d: 4a:34:f0:13:1f:27:c1:c8:db:f5:7a:8a:9a:89:1c: c1:e8:49:85:73:08:e7:e5:75:85:c2:8b:fa:b1:32: c6:c3:85:ec:e5:88:0f:2d:39:3d:4e:96:8a:47:d0: 7d:0e:c8:86:f1:31:7d:de:9a:75:7f:38:ef:06:5a: ed:32:a4:af:bd:64:8b:c0:0d:39:3e:06:a8:4f:f6: 8b:c0:04:37:f6:7a:53:68:eb:b9:ea:92:09:89:0c: 63:5a:02:2b:27:42:bb:82:18:b9:98:cd:98:11:8b: ae:74:30:49:e3:a9:01:b4:79:40:c0:90:c4:b2:dc: 7e:f3:f7:57:0b:5e:a5:09:d4:b8:e2:a1:1b:90:8f: 9f:99:db:19:b3:42:2a:10:40:8e:f3:90:07:fd:cc: 4d:11 Exponent: 65537 (0x10001)	/kCEWHjOYw9nll76 34JmsFlijqN727A0XQFg6InDFh9Yb0RcZVc6uiNz+w3TE xvJ95Yzdp1uBG5IbGP2 8Zj1DIDPGxVphSVZiHbJDxFb3WY2k7fLvZUwrVZOv2i+p et8gy2F86NGeRbFSo4r wwIDAQAB -----END PUBLIC KEY----- RSA Public-Key: (2048 bit) Modulus: 00:c3:08:44:e1:2b:eb:64:ea:d5:31:65:5c:71:06: cc:b4:76:e1:a8:8e:a7:d9:d2:c6:36:9c:fd:82:e3: d2:b6:da:47:3a:a6:e1:8c:8a:d6:15:11:a1:fb:0b: bc:af:37:ec:3a:86:94:4c:e4:8e:c4:74:50:20:83: dc:92:b9:8f:22:26:da:f7:ec:9b:b8:16:1b:a3:05: 6d:44:16:3a:e4:81:7f:88:63:1b:8e:3e:dc:be:40: 2f:80:e9:b6:35:57:8b:00:11:ae:32:92:44:70:30: 5f:bf:a4:33:c1:11:b6:e9:8e:6d:0f:b3:35:e0:fc: 57:86:d1:03:a6:73:63:2f:f9:af:8a:2b:51:f8:bc: 4b:1a:1b:07:a7:bf:65:35:5d:bb:ce:26:89:fe:40: 84:58:78:ce:63:0f:67:96:5e:fa:df:82:66:b0:59: 62:8e:a3:7b:db:b0:34:5d:01:60:e8:89:c3:16:1f: 58:6f:44:5c:65:57:3a:ba:23:73:fb:0d:d3:13:1b: c9:f7:96:33:76:9d:6e:04:6e:48:6c:63:f6:f1:98: f5:0c:80:cf:1b:15:69:85:25:59:88:76:c9:0f:11: 5b:dd:66:36:93:b7:cb:bd:95:30:ad:56:4e:bf:68: be:a5:eb:7c:83:2d:85:f3:a3:46:79:16:c5:4a:8e: 2b:c3 Exponent: 65537 (0x10001)
Sent *Alice01-public-key.pem* to Bob over email	Sent *Bob29-public-key.pem* to Alice over email
Created *SA01.key* with the following info $ cat SA01.key -aes-256-cbc, 1000, Alice@4	Created *SB29.key* with the following info $ cat SB29.key -aes-256-cbc, 1200, khalid@007
Mechanism for proving Authenticity and Integrity: As SA01.key is small we can directly compute the signature without any digest. Computing Signature for *SA01.key* $ openssl pkeyutl -sign -in SA01.key -out Alice01-signature.key -inkey Alice01-private-key.pem \\ Enter pass phrase for Alice01-private-key.pem:edith@4	**Mechanism for proving Authenticity and Integrity:** As SB29.key is small we can directly compute the signature without any digest. Computing Signature for *SB29.key* $ openssl pkeyutl -sign -in SB29.key -out Bob29-signature.key -inkey Bob29-private-key.pem \\ Enter pass phrase for Bob29-private-key.pem:khalid
Sent *SA01.key*, *Alice01-signature.key* to Bob	Sent *SB29.key*, *Bob29-signature.key* to Alice

Verifying Authenticity and Integrity of Bob: $ openssl pkeyutl -verify -sigfile Bob29-signature.key -in SB29.key -inkey Bob29-public-key.pem -pubin Signature Verified Successfully This indicates that the file signatured with Bob's private key can be extracted with Bob's public key(Authenticity) and the whole file matches with the sent SB29.key file(integrity).	**Verifying Authenticity and Integrity of Alice:** $ openssl pkeyutl -verify -sigfile Alice01-signature.key -in SA01.key -inkey Alice01-public-key.pem -pubin Signature Verified Successfully This indicates that the file signatured with Alice's private key can be extracted with Alice's public key(Authenticity) and the whole file matches with the sent SA01.key file(integrity).
Encrypting a large file with parameters given in SA01.key $ openssl enc -aes-256-cbc -e -iter 1000 -salt -in AliceOriginalFile.pdf -out AliceEncFile.pdf \\ enter aes-256-cbc encryption password:Alice@4 \\ Verifying - enter aes-256-cbc encryption password:Alice@4	**Encrypting a large file with parameters given in SB29.key** $ openssl enc -aes-256-cbc -e -iter 1200 -salt -in BobOriginalFile.pdf -out BobEncFile.pdf \\ enter aes-256-cbc encryption password:khalid@007 \\ Verifying - enter aes-256-cbc encryption password:khalid@007
Sent *AliceEncFile.pdf* to Bob	Sent *BobEncFile.pdf* to Alice
Decrypting the encrypted file received from Bob using SB29.key $ openssl enc -aes-256-cbc -d -iter 1200 -in BobEncFile.pdf -out BobDecFile.pdf \\ enter aes-256-cbc decryption password:khalid@007	**Decrypting the encrypted file received from Alice using SA01.key** $ openssl enc -aes-256-cbc -d -iter 1000 -in AliceEncFile.pdf -out AliceDecFile.pdf \\ enter aes-256-cbc decryption password:Alice@4

PART B

Charlie	Bob
Generating a Self-Signed Certificate for Charlie $ openssl req -newkey rsa:2048 -nodes -keyout Charlie-private-key.pem -x509 -days 365 -out Charlie-CA.crt Generating a RSA private key .....+++++ .....+++++ writing new private key to 'Charlie-private-key.pem' ----- 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]:IN State or Province Name (full name) [Some-State]:AP Locality Name (eg, city) []:Vijayawada Organization Name (eg, company) [Internet Widgits Pty Ltd]:Avengers Organizational Unit Name (eg, section) []:Marvel Common Name (e.g. server FQDN or YOUR name) []:Edith Email Address []:edithdeath4@gmail.com	Generating a CSR request with the Bob's private key $ openssl req -key Bob29-private-key.pem -new -out Bob29-browser.csr You are about to be asked to enter information that will be incorporated into your certificate request. What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank For some fields there will be a default value, If you enter '.', the field will be left blank. ----- Country Name (2 letter code) [AU]: IN State or Province Name (full name) [Some-State]: TS Locality Name (eg, city) []: palakurthy Organization Name (eg, company) [Internet Widgits Pty Ltd]: E-corp Organizational Unit Name (eg, section) []: meh Common Name (e.g. server FQDN or YOUR name) []: skipped Email Address []: random_email@email.com Please enter the following 'extra' attributes to be sent with your certificate request A challenge password []: An optional company name []:
Viewing Self-signed certificate of Charlie $ openssl x509 -in Charlie-CA.crt -text Certificate: Data: Version: 3 (0x2) Serial Number: 5b:4e:c9:39:3e:e0:96:23:df:1a:e8:c4:81:45:ca:62:03:ab:2e:74 Signature Algorithm: sha256WithRSAEncryption Issuer: C = IN, ST = AP, L = Vijayawada, O = Avengers, OU = Marvel, CN = Edith, emailAddress =	Viewing CSR generated by Bob $ openssl req -in Bob29-browser.csr -text Certificate Request: Data: Version: 1 (0x0) Subject: C = IN, ST = TS, L = palakurthy, O = E-corp, OU = meh, CN = skipped, emailAddress = random_email@email.com Subject Public Key Info: Public Key Algorithm: rsaEncryption RSA Public-Key: (2048 bit) Modulus:

edithdeath4@gmail.com
Validity
Not Before: Feb 7 15:58:46 2021 GMT
Not After : Feb 7 15:58:46 2022 GMT
Subject: C = IN, ST = AP, L = Vijayawada, O = Avengers, OU = Marvel, CN = Edith, emailAddress = edithdeath4@gmail.com
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public-Key: (2048 bit)
Modulus:

00:b7:d7:61:8c:6b:43:19:ed:f0:8d:7d:91:02:12:

52:86:ac:fc:33:69:e6:67:7a:58:31:a5:7c:34:6e:

44:bf:f6:73:bb:74:82:3e:c4:7a:7d:ae:8d:d3:ec:

f0:0d:8e:62:63:5f:cc:8a:40:4c:b6:23:d7:50:3b:

ea:89:2e:91:c8:b8:33:99:ec:ad:b9:20:11:02:6f:

7f:ad:d9:d2:f9:34:d4:34:9a:86:4c:47:1a:e6:c2:
5a:fe:09:08:0b:8f:40:8f:c1:f7:3d:79:29:e1:d9:
eb:f0:c6:e5:36:cb:c7:90:ce:f6:18:d3:aa:35:e1:

ed:34:ee:43:ac:b8:5f:d7:1e:9b:3d:8f:39:8a:90:

d5:a5:37:81:89:2f:46:a2:65:d6:66:c7:4d:34:50:
e0:d1:cd:f0:63:46:98:ad:eb:7c:29:74:0f:11:ca:

46:af:63:43:e6:a3:2a:ff:06:ed:80:ce:e9:2a:e3:
68:51:ec:14:9c:b3:09:e7:92:96:f1:3b:4f:c1:b6:

dc:a3:15:65:31:a2:73:d2:ab:a1:38:dc:e3:48:4b:
3b:2f:f4:04:33:a1:d4:8d:be:91:9b:11:98:e3:10:

f1:d4:0e:5c:88:5d:d6:8b:66:96:cd:de:23:5f:d7:
fb:c3:7c:8b:f5:bd:f4:ba:01:4f:da:23:e4:5e:20:
be:fd
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Subject Key Identifier:

63:F6:7C:0E:88:BA:7F:7A:BE:5F:32:A2:56:30:77:EE:EF:2F:93:71
X509v3 Authority Key Identifier:

00:cd:36:ec:af:f0:8a:8e:00:38:55:55:fc:e2:ad:

5e:e4:38:1f:31:73:27:b3:6c:ce:74:a1:01:6c:5d:

62:a6:25:f8:d3:80:ab:a2:34:bf:7b:2d:4f:d8:05:

74:ac:22:2e:30:f3:c0:45:54:60:3b:2b:c9:9d:f1:

bd:6a:1f:9b:81:c6:8e:86:db:05:ca:81:2b:6a:6a:

13:17:5b:cb:5e:22:09:2d:db:eb:01:b1:09:fb:c2:

07:5b:86:17:ef:bf:9b:03:3f:52:0b:60:f9:13:93:

ad:f4:f3:9b:23:22:fa:d7:70:b9:57:60:ba:3d:24:
92:19:b3:f7:1b:7d:f4:52:31:69:ff:1d:65:c1:de:

0f:46:a1:c9:f5:1f:17:a6:20:bd:59:6f:28:22:5d:

11:f5:d2:85:40:3b:99:0a:73:1c:3c:67:f4:c6:b9:

70:14:c0:d4:b2:e6:8c:10:eb:bc:d5:0c:29:3c:b9:

19:26:50:a5:3d:0d:59:88:22:bc:07:72:b0:09:23:

79:3b:54:8b:d2:6b:67:33:3b:71:0e:6d:52:2f:c8:

7b:67:bb:28:89:5e:49:90:00:71:17:e5:84:58:6f:

40:3c:2b:a5:7d:75:60:e9:43:26:11:a1:83:b5:20:

99:98:62:66:e5:42:5a:a4:bf:06:09:67:ed:83:d9:
c2:c9
Exponent: 65537 (0x10001)
Attributes:
a0:00
Signature Algorithm: sha256WithRSAEncryption

1d:29:88:03:32:00:82:e4:2e:21:58:50:07:82:bd:4d:a5:f5:

1c:49:41:ab:46:a5:b1:a7:2f:a5:70:31:47:c7:f3:4f:4d:b3:

9e:13:f0:a7:37:48:a6:9f:64:ed:45:11:9b:b9:94:5e:47:1b:

5c:c7:a5:88:bb:3d:a7:c8:53:10:66:f1:e4:ed:09:f4:03:fa:

keyid:63:F6:7C:0E:88:BA:7F:7A:BE:5F:32:A2:56:30:77:E
E:EF:2F:93:71

X509v3 Basic Constraints: critical
CA:TRUE
Signature Algorithm: sha256WithRSAEncryption

a8:7c:f0:9c:23:9a:e4:97:97:1f:ca:37:01:0d:0b:9e:a2:35:

56:7f:97:fc:c1:5b:11:d3:d9:0d:a6:37:76:ae:cd:3a:77:43:

fb:bd:85:ea:b6:41:4e:62:36:d3:c8:a8:be:12:5e:8f:1c:c2:

b4:ed:c1:0d:01:b8:94:30:de:b4:ce:82:1e:f1:52:f3:1d:db:

b1:79:4f:4d:28:6f:95:76:6a:90:21:cd:21:e6:5e:3a:04:22:

97:07:2c:9c:0d:7e:5f:75:3a:d2:99:b8:b6:4b:6c:01:21:5e:

c3:de:6d:9a:a9:44:8c:fe:ed:b9:c6:63:ae:13:2b:fb:11:09:

f8:c2:f6:e8:36:51:b2:5f:0c:92:5b:20:43:76:10:85:f5:d8:

4b:5e:93:e8:20:9b:ca:a4:82:10:e9:ab:a8:cd:ca:6e:7a:39:

24:3a:fc:24:a3:41:16:0c:d5:91:d3:cd:ea:c7:df:5f:47:5b:

cd:94:b0:03:be:8c:b2:fa:a0:cd:b4:60:a7:9a:32:01:d8:4f:

07:ce:8e:83:fe:07:e6:31:18:fb:fb:a1:f8:90:8c:b6:5a:42:

d8:b7:95:f5:bc:7f:72:f1:ce:f4:92:1d:b6:f2:c2:58:d1:2a:

5b:10:14:7b:93:1e:e2:62:34:bb:dc:d0:df:c7:81:a1:55:15:
58:3d:fa:98
-----BEGIN CERTIFICATE-----
MIID9TCCAt2gAwIBAgIUW07JOT7gliPfGujEgUXKYgOr
LnQwDQYJKoZIhvcNAQEL
BQAwgYkxCzAJBgNVBAYTAklOMQswCQYDVQQIDAJ
BUDETMBEGA1UEBwwKVmlqYXlh
d2FkYTERMA8GA1UECgwIQXZlbmdlcnMxDzANBgNVB
AsMBk1hcnZlbDEOMAwGA1UE
AwwFRWRpdGgxJDAiBgkqhkiG9w0BCQEWFWVkaXRo
ZGVhdGg0QGdtYWlsLmNvbTAe
Fw0yMTAyMDcxNTU4NDZaFw0yMjAyMDcxNTU4NDZ
aMIGJMQswCQYDVQQGEwJJTjEL
4a:05:94:c0:7c:ae:05:7e:4d:ca:f7:be:2a:de:af:8c:0d:ec:

51:5f:3f:5f:ae:58:4c:56:f8:66:50:2e:15:6b:ad:b1:69:17:

ee:06:61:99:91:f0:59:b3:96:98:dc:1c:60:b1:b2:c7:71:d5:

8b:54:41:ef:b2:a2:e9:90:52:d1:0c:b1:bc:f6:e4:9d:61:ac:

30:3b:02:33:00:8f:9b:6a:ee:1e:21:08:26:8a:4b:32:97:7a:

87:1e:bc:b2:14:c2:15:f2:24:29:1e:80:f8:07:bb:75:2b:4e:

db:b7:c1:f8:b4:8b:1e:78:4c:bb:c6:57:ea:b0:f6:a0:b5:d7:

86:60:21:c5:98:ba:6e:13:c7:14:9b:e9:58:b1:b8:90:c6:85:

de:8f:8c:8c:b7:59:dd:74:08:c0:61:02:36:52:87:b2:1a:c2:

dd:7f:ed:ae:c7:4c:4b:5a:dc:6c:3e:35:72:0b:19:d3:7a:5c:
5e:18:56:07
-----BEGIN CERTIFICATE REQUEST-----
MIICzTCCAbUCAQAwgYcxCzAJBgNVBAYTAklOMQsw
CQYDVQQIDAJUUzETMBEGA1UE
BwwKcGFsYWt1cnRoeTEPMA0GA1UECgwGRS1jb3JwM
QwwCgYDVQQLDANtZWgxEDAO
BgNVBAMMB3NraXBwZWQxJTAjBgkqhkiG9w0BCQEW
FnJhbmRvbV9lbWFpbEBlbWFp
bC5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwg
gEKAoIBAQDNNuyv8IqOADhV
VfzirV7kOB8xcyezbM50oQFsXWKmJfjTgKuiNL97LU/YB
XSsIi4w88BFVGA7K8md
8b1qH5uBxo6G2wXKgStqahMXW8teIgkt2+sBsQn7wgd
bhhfvv5sDP1ILYPkTk630
85sjlvrXcLIXYLo9JJIZs/cbffRSMWn/HWXB3g9Gocn1Hxe
mIL1ZbygiXRH10oVA
O5kKcxw8Z/TGuXAUwNSy5owQ67zVDCk8uRkmUKU9
DVmIIrwHcrAJl3k7VIvSa2cz
O3EObVIvyHtnuyiJXkmQAHEX5YRYb0A8K6V9dWDpQ
yYRoYO1IJmYYmblQIqkvwYJ
Z+2D2cLJAgMBAAGgADANBgkqhkiG9w0BAQsFAAOC
AQEAHSmIAzIAguQuIVhQB4K9
TaX1HEIBq0aIsacvpXAxR8fzT02znhPwpzdIpp9k7UURm
7mUXkcbXMeliLs9p8hT
EGbx5O0J9AP6SgWUwHyuBX5Nyve+Kt6vjA3sUV8/X6
5YTFb4ZIAuFWutsWkX7gZh
mZHwWbOWmNwcYLGyx3HVi1RB77Ki6ZBS0QyxvPbkn
WGsMDsCMwCPm2ruHiElJopL

MAkGA1UECAwCQVAxEzARBgNVBAcMClZpamF5YXdhZGExETAPBgNVBAoMCEF2ZW5nZXJzMQ8wDQYDVQQLDAZNYXJ2ZWwxDjAMBgNVBAMMBUVkaXRoMSQwIgYJKoZIhvcNAQkBFhVlZGl0aGRlYXRoNEBnbWFpbC5jb20wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC312GMa0MZ7fCNfZECEIKGrPwzaeZneIgxpXw0bkS/9nO7dII+xHp9ro3T7PANjmJjX8yKQEy2I9dQO+qJLpHIuDOZ7K25IBECb3+t2dL5NNQ0moZMRxrmwlr+CQgLj0CPwfc9eSnh2evwxuU2y8eQzvYY06o14e007kOsuF/XHps9jzmKkNWIN4GJL0aiZdZmx000UODRzfBjRpit63wpdA8RykavY0Pmoyr/Bu2Azukq42hR7BScswnnkpbxO0/BttyjFWUxonPSq6E43ONISzsv9AQzodSNvpGbEZjjEPHUDlyIXdaLZpbN3iNf1/vDflv1vfS6AU/al+ReIL79AgMBAAGjUzBRMB0GA1UdDgQWBBRj9nwOiLp/er5fMqJWMHfu7y+TcTAfBgNVHSMEGDAWgBRj9nwOiLp/er5fMqJWMHfu7y+TcTAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQCofPCcI5rkI5cfyjcBDQueojVWf5f8wVsR09kNpjd2rs06d0P7vYXqtkFOYjbTyKi+El6PHMK07cENAbiUMN60zole8VLzHduxeU9NKG+VdmqQIc0h5I46BCKXByycDX5fdTrSmbi2S2wBIV7D3m2aqUSM/u25xmOuEyv7EQn4wvboNlGyXwySWyBDdhCF9dhLXpPoIJvKpIlQ6auozcpuejkkOvwko0EWDNWR083qx99fR1vNlLADvoyy+qDNtGCnmjlB2E8Hzo6D/gfmMRj7+6H4kIy2WkLYt5X1vH9y8c70kh228sJY0SpbEBR7kx7iYjS73NDfx4GhVRVYPfqY -----END CERTIFICATE-----	Mpd6hx68shTCFflkKR6A+Ae7dStO27fB+LSLHnhMu8ZX6rD2oLXXhmAhxZi6bhPHFJvpWLG4kMaF3o+MjLdZ3XQIwGECNlKHshrC3X/trsdMS1rcbD41cgsZ03pcXhhW Bw== -----END CERTIFICATE REQUEST-----
Creating End User cert for *Bob29-browser.csr* by Charlie $ openssl x509 -req -sha256 -days 365 -in Bob29-browser.csr -CAkey Charlie-private-key.pem -CA Charlie-CA.crt -out Bob29-browser.crt -CAcreateserial Signature ok subject=C = IN, ST = TS, L = palakurthy, O = E-corp, OU = meh, CN = skipped, emailAddress = random_email@email.com Getting CA Private Key	Sent Bob29-browser.csr to Charlie for requesting of End User Certificate

Sent Bob29-browser.crt to Bob Sent Charlie-CA.crt to Alice	Sent Bob29-browser.crt to Alice

Alice verifying whether Bob's certificate is valid or not:

$ openssl verify -verbose -CAfile Charlie-CA.crt Bob29-browser.crt

Bob29-browser.crt: OK