CS6903: Network Security

Assignment 1: ABCs of Digital Certificates

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

Date: 06/02/2021

Signature: CS18BTECH11001

Field Name	Subject (CN) of certificate holder (website)	Subject (CN) of certificate holder (intermediate)	Subject (CN) of certificate holder (root)	Remarks/observations
Issuer	GTS CA 1O1	GlobalSign	GlobalSign	None
Version No.	3	3	3	All the certificates are using version 3
Signature Algo	SHA-256 with RSA Encryption	SHA-256 with RSA Encryption	SHA1 with RSA Encryption	As root certification is created a way long it uses older SHA1 encryption
Size of digest	256	256	160	SHA256 has a digest size of 256 and hence providing higher encryption than SHA1 with digest size as 160
Signature Value	B6 12 FC 6E B0 AE 1C EB BC 16 1C 06 83 DD A9 D3 62 24 6F F5 C7 05 8A 20 AC 0D 5E ED 8C 33 B8 F5 FC D0 9F 1C DF C4 95 89 12 4D 48 D9 D0 18 7D 9A C7 AF 1A E3 75 C3 E0 A9 B0 85 4E 02 93 75 15 32 64 2C 89 94 D9 B3 D4 8A 89 B7 77 8F 86 41 EE D7 B8 D2 87 AB CE 34 FC 05 83 9E 92 84 87 64 51 74 AD 4B 99 21 67 91 B5 BF CD F6 49 90 16 2E F1 84 86	1A 80 3E 36 79 FB F3 2E A9 46 37 7D 5E 54 16 35 AE C7 4E 08 99 FE BD D1 34 69 26 52 66 07 3D 0A BA 49 CB 62 F4 F1 1A 8E FC 11 4F 68 96 4C 74 2B D3 67 DE B2 A3 AA 05 8D 84 4D 4C 20 65 0F A5 96 DA 0D 16 F8 6C 3B DB 6F 04 23 88 6B 3A 6C C1 60 BD 68 9F 71 8E EE 2D 58 34 07 F0 D5 54 E9 86 59 FD 7B 5E 0D 21 94 F5 8C C9 A8 F8 D8 F2 AD CC 0F 1A F3 9A	99 81 53 87 1C 68 97 86 91 EC E0 4A B8 44 0B AB 81 AC 27 4F D6 C1 B8 1C 43 78 B3 0C 9A FC EA 2C 3C 6E 61 1B 4D 4B 29 F5 9F 05 1D 26 C1 B8 E9 83 00 62 45 B6 A9 08 93 B9 A9 33 4B 18 9A C2 F8 87 88 4E DB DD 71 34 1A C1 54 DA 46 3F E0 D3 2A AB 6D 54 22 F5 3A 62 CD 20 6F BA 29 89 D7 DD 91 EE D3 5C A2 3E A1 5B 41 F5 DF E5 64 43 2D E9 D5 39 AB D2 A2	None

	F8 C0 33 2A BA DD 95 B5 8D AB EE 17 C6 F7 BA 70 37 F0 4D 78 27 70 61 41 B0 45 1E 60 16 31 4B BD 31 09 9B 88 DB 79 10 5B 99 74 1C 90 87 DD DA 06 75 83 77 E5 7C C0 8F EB B6 AE F1 43 FB E6 BC 2C A0 B0 4C 82 7C 23 36 FB 14 75 08 F1 8E 27 43 0B FF 36 8F C9 D5 0C F1 B9 F1 EE 29 34 ED EB A2 73 AF AC 45 F4 A5 4F A5 6B DB 37 1E 8F 2B 12 E0 3A B8 B3 B9 A0 39 75 DE E5 00 F2 0B 4D F7 B2 D0 D7 DF 7D 90 89 46 49 53 E1 AE 65 55 51 7F 6D DA	A7 A9 04 27 F9 A3 C9 B0 FF 02 78 6B 61 BA C7 35 2B E8 56 FA 4F C3 1C 0C ED B6 3C B4 4B EA ED CC E1 3C EC DC 0D 8C D6 3E 9B CA 42 58 8B CC 16 21 17 40 BC A2 D6 66 EF DA C4 15 5B CD 89 AA 9B 09 26 E7 32 D2 0D 6E 67 20 02 5B 10 B0 90 09 9C 0C 1F 9E AD D8 3B EA A1 FC 6C E8 10 5C 08 52 19 51 2A 71 BB AC 7A B5 DD 15 ED 2B C9 08 2A 2C 8A B4 A6 21 AB 63 FF D7 52 49 50 D0 89 B7 AD F2 AF FB 50 AE 2F E1 95 0D F3 46 AD 9D 9C F5 CA	DF B7 8B D0 C0 80 19 1C 45 C0 2D 8C E8 F8 2D A4 74 56 49 C5 05 B5 4F 15 DE 6E 44 78 39 87 A8 7E BB F3 79 18 91 BB F4 6F 9D C1 F0 8C 35 8C 5D 01 FB C3 6D B9 EF 44 6D 79 46 31 7E 0A FE A9 82 C1 FF EF AB 6E 20 C4 50 C9 5F 9D 4D 9B 17 8C 0C E5 01 C9 A0 41 6A 73 53 FA A5 50 B4 6E 25 0F FB 4C 18 F4 FD 52 D9 8E 69 B1 E8 11 0F DE 88 D8 FB 1D 49 F7 AA DE 95 CF 20 78 C2 60 12 DB 25 40 8C 6A FC 7E 42 38 40 64 12 F7 9E 81 E1 93 2E	
Validity period	19/1/2021, 1:27:09 pm (IST) - 13/4/2021, 1:27:08 pm (IST)	15/6/2017, 5:30:42 am (IST) - 15/12/2021, 5:30:42 am (IST)	15/12/2006, 1:30:00 pm (IST) - 15/12/2021, 1:30:00 pm (IST)	This can be used as one of the parameter in checking whether the certificate is valid or not
Subject (CN)	*.google.com	GTS CA 1O1	GlobalSign	None
Certificate type: DV, IV, OV or EV? Tell also how you are able to determine the type!	OV Able to see the following general details CN = *.google.com O = Google LLC L = Mountain View	OV Able to see the following general details CN = GTS CA 1O1 O = Google Trust Services C = US	OV Able to see the following general details CN = GlobalSign O = GlobalSign OU = GlobalSign Root CA - R2	As the certificates subject details about the organisation and thus providing most trust these are classified as OU

	ST = California C = US			
Subject Alternative Name (SAN), if any	*.android.com, *.bdn.dev (approx 73 SANs)	----	----	The end user is having many SANs whereas the intermediate and the root has a single domain certificate.
Certificate category: Single domain, wildcard or SAN/UCC cert?	SAN/UCC cert	Single domain cert	Single domain cert	The end user is having many SANs whereas the intermediate and the root has a single domain certificate.
Public Key Info like key algo, key length, public exponent (e) in case of RSA	Algo: Elliptic Curve Length: 256	Algo: RSA Length: 2048 Exponent: 65537	Algo: RSA Length: 2048 Exponent: 65537	As the end user performance should be much greater than the root CA, it uses an Elliptic curve with less number of bits than RSA.
Public key or modulus (n) in case of RSA	Public key: 04:53:D3:05:3C:10:D8:CC:8D:06:A0:1C:02:17:1E:8C:2D:91:B3:55:CC:18:81:12:94:3A:21:7E:DC:2F:E6:0E:35:92:F3:29:40:45:73:E1:24:C0:77:91:7D:CF:31:9F:14:A6:A2:C3:E4:33:EE:69:5D:60:A7:E9:BA:38:83:AA:5B	Modulus: D0:18:CF:45:D4:8B:CD:D3:9C:E4:40:EF:7E:B4:DD:69:21:1B:C9:CF:3C:8E:4C:75:B9:0F:31:19:84:3D:9E:3C:29:EF:50:0D:10:93:6F:05:80:80:9F:2A:A0:BD:12:4B:02:E1:3D:9F:58:16:24:FE:30:9F:0B:74:77:55:93:1D:4B:F7:4D:E1:92:82:10:F6:51:AC:0C:C3:B2:22:94:0F:34:6B:98:10:49:E7:0B:9D:83:39:DD:20:C6:1C:2D:EF:D1:18:61:65:E7:23:83:20:	Modulus: A6:CF:24:0E:BE:2E:6F:28:99:45:42:C4:AB:3E:21:54:9B:0B:D3:7F:84:70:FA:12:B3:CB:BF:87:5F:C6:7F:86:D3:B2:30:5C:D6:FD:AD:F1:7B:DC:E5:F8:60:96:09:92:10:F5:D0:53:DE:FB:7B:7E:73:88:AC:52:88:7B:4A:A6:CA:49:A6:5E:A8:A7:8C:5A:11:BC:7A:82:EB:BE:8C:E9:B3:AC:96:25:07:97:4A:99:2A:07:2F:B4:1E:77:BF:8A:0F:B5:02:7C:1B:96:B8:	None

		A8:23:12:FF:D2:24:7F:D4:2F:E7:44:6A:5B:4D:D7:50:66:B0:AF:9E:42:63:05:FB:E0:1C:C4:63:61:AF:9F:6A:33:FF:62:97:BD:48:D9:D3:7C:14:67:DC:75:DC:2E:69:E8:F8:6D:78:69:D0:B7:10:05:B8:F1:31:C2:3B:24:FD:1A:33:74:F8:23:E0:EC:6B:19:8A:16:C6:E3:CD:A4:CD:0B:DB:B3:A4:59:60:38:88:3B:AD:1D:B9:C6:8C:A7:53:1B:FC:BC:D9:A4:AB:BC:DD:3C:61:D7:93:15:98:EE:81:BD:8F:E2:64:47:20:40:06:4E:D7:AC:97:E8:B9:C0:59:12:A1:49:25:23:E4:ED:70:34:2C:A5:B4:63:7C:F9:A3:3D:83:D1:CD:6D:24:AC:07	C5:B9:3A:2C:BC:D6:12:B9:EB:59:7D:E2:D0:06:86:5F:5E:49:6A:B5:39:5E:88:34:EC:BC:78:0C:08:98:84:6C:A8:CD:4B:B4:A0:7D:0C:79:4D:F0:B8:2D:CB:21:CA:D5:6C:5B:7D:E1:A0:29:84:A1:F9:D3:94:49:CB:24:62:91:20:BC:DD:0B:D5:D9:CC:F9:EA:27:0A:2B:73:91:C6:9D:1B:AC:C8:CB:E8:E0:A0:F4:2F:90:8B:4D:FB:B0:36:1B:F6:19:7A:85:E0:6D:F2:61:13:88:5C:9F:E0:93:0A:51:97:8A:5A:CE:AF:AB:D5:F7:AA:09:AA:60:BD:DC:D9:5F:DF:72:A9:60:13:5E:00:01:C9:4A:FA:3F:A4:EA:07:03:21:02:8E:82:CA:03:C2:9B:8F	
Key usages	Digital Signature, Server Authentication	Digital Signature, Revocation list Signature, Server Authentication, Client Authentication	Certificate Signing, CRL Signing	The usages are different for root, intermediate and end user. According to my understanding as root CA uses self signing and intermediate signing it uses its certificate for online client Authentication where as intermediate tries to connect to both root and

				end user it uses this certificate for both server and client authentication where as end user only connect to intermediate it only uses Server Authentication.
Basic constraints	CA: No, Max Path Length: Unlimited	CA: Yes, Max Path Length: 0	CA: Yes, Max Path Length: 0	CA: No for end user indicates that it cannot provide certificate for others
Name constraints, how these are useful?	CA: No indicates that it cannot provide certificate to other authorities	Max Path length:0 indicates it that there must not be any CA certificate under this CA certificate	CA: Yes indicates that it can provide certificates to its clients	None
Size of the certificate	3,414 bytes	1,574 bytes	1,376 bytes	None
Any other parameter? Finger print	95 BD B9 91 B0 24 D1 74 96 23 0E 7D F3 E2 91 C5 90 43 23 CB 0F 16 BC 2E BE C5 D0 D6 A6 5A B3 25	95 C0 74 E3 59 02 A1 4A BD 9D 19 AF B6 E7 F8 0E 66 9F F8 E2 36 32 70 53 9D 96 36 13 F0 4A AA 21	CA 42 DD 41 74 5F D0 B8 1E B9 02 36 2C F9 D8 BF 71 9D A1 BD 1B 1E FC 94 6F 5B 4C 99 F4 2C 1B 9E	The fingerprint is the hash of the entire certificate. The values mentioned here are SHA256 fingerprint hashes.

Answer the following queries after filling out the above table:

1. Which certificate type (DV/OV/IV/EV) is more trustable and expensive?

A. As EV certificates provide the maximum amount of trust to visitors. So it requires more effort by the CA to validate. And thus it is considered to be more trustable and expensive.

2. What is the role of Subject Alternative Name (SAN) field?
A. SAN allows you to specify additional hostnames for a single SSL certificate.

3. Why are key usages and basic constraints different for root, intermediate and end certificates?
A. As root is present in the higher level of the hierarchy and as it uses the self signed certificates and provides the access to the intermediates which further certifies the end user, the usages for these levels of hierarchy will be different as explained in the remarks section in the above table.

4. Why do RSA key lengths increase over the years? Why ECDSA is being preferred over RSA now-a-days?
A. The increase in use over RSA keys and the academic successes in breaking bit strengths have made RSA to use longer keys to provide a safe level of encryption protection and thus slowing down its performance. As ECDSA uses shorter key length with a stronger hash algorithm that offers a better performance than RSA, it is preferred now-a-days.

5. What are pros and cons of pre-loading root and intermediate certificates in the root stores of browsers and OSes?
A. Pros:
When we are purchasing/downloading a new OS or browser without any preloaded certificates then it can't be trustable whether a particular website is secure or not. But using a preloaded root or intermediate certificate will ensure this clause.
Cons:
If any of the root or intermediate certificates is closed due to some compromisation issues, then the browsers or OS which has pre-loaded these certificates can be attacked by an evil trudy which captured the charge for these certificates.

6. Why are root CAs kept offline?
A. If the root CAs are connected to any of the company's network, there is a high chance for attackers to capture the private key of the root CA. if the root CA is compromised, all the intermediaries and end user certificates issued by it will get compromised. Hence to avoid this vulnerability, the root CAs will be kept offline and to issue certificates it uses removable media such as USB, floppy disks, CDs which can then be physically transported to the intermediate CAs.

PART-B

1. You have received a digital certificate of a firm M over email. How do you verify whether the certificate is valid without using any of online tools or browsers? Write a psuedo-code of your verifier function named myCertChecker( ) and explain how it works by picking the chain of trust of an end-user cert in PART-A of this assignment.

A.
```
Bool myCertChecker(cert M) {
        sys_time = system.getTime() // present time in IST
        if(sys_time <= M.NotValidBefore || sys_time >= M.NotValidAfter)
            Return invalid
        myCertChecker(M.Issuer)      // check the certificate of issuer(intermediate & root)

        return valid
}
```
Explanation:
This first checks whether the given certificate is valid or not using the time of expiry mentioned in the certificate. Then it goes to the next higher level(intermediate or root) and then checks the same. This is a recursive function which checks until the root CA and if all the requirements are satisfied then it returns valid else it returns invalid.

2. Consider the scenario in which evil Trudy has used the digital certificate of firm M to launch her own web server. Does your function myCertChecker( ) returns valid or invalid for this when someone tries to access Trudy's website from a browser like Chrome/Edge/Firefox?

A. My function will not check the user preference and considers on the validity of the certificate. So it will return valid even it the evil Trudy has used the digital certificate of firm M.

7-zip:

I have used a normal inbuilt compressor to compress all the files in the .7z format. Then I have used the openssl command explained in the tutorial to encrypt the file. The command is given below:

```
$ openssl enc -aes-256-cbc -e -in DCAsg.7z -out DCAsg-CS18BTECH11001.7z
enter aes-256-cbc encryption password: CS18BTECH11001
Verifying - enter aes-256-cbc encryption password: CS18BTECH11001
```

The password entered is hashed to maintain integrity and uses aes-256 symmetric encryption for authenticity.