Assignment 4: ABCs of Digital Certificates

PART-A: Comparison of Digital Certificates in the chain of trust of a popular website

- Website: http://www.tiktok.com

Field Name	*.www.tiktok.com	RapidSSL ECC CA 2018	DigiCert Global Root CA	Remarks/observations
Issuer	RapidSSL ECC CA 2018	DigiCert Global Root CA	DigiCert Global Root CA	Depicts who authorizes whom and breaking the circular definition by the root who authorizes itself
Version No.	Version 3	Version 3	Version 3	Adding some more extension like keyUsage, extendedKeyUsage, etc while signing the certificate results into Version 3 certificate
Signature Algo	X9.62 ECDSA Signature with SHA-256	PKCS #1 SHA-256 With RSA Encryption	PKCS #1 SHA-1 With RSA Encryption	Indicates the underlying algorithm using which the certificate was hashed and encrypted to form its signature value
Size of digest that is signed to generate Cert Sign	256 bits	256 bits	160 bits	Generated Hash Length
Size of Cert	568 bits	2048 bits	568 bits	Using the ASA.1

Signature				**Javascript Decoder we can dig into the structure of the digital certificate which also includes the size of certificate signature**
Validity period	**09-11-2023 to 10-12-2024**	**06-11-2017 to 06-11-2027**	**10-11-2006 to 10-11-2031**	**As we move bottom up in the hierarchy of CA's the validity period of certificate increase, whose reason is self explanatory**
Is Subject field (CN), FQDN?	**Yes**	**No**	**No**	**FQDN stands for Fully Qualified Domain Name or can be aliased as Absolute Domain Name which specifies exact location in tree hierarchy of DNS**
Certificate type: DV, IV, OV or EV? Tell also how you are able to determine the type!	**DV**	**OV**	**EV**	**Using various steps we can detect the same, some of which are checking the subject field under the detail tab, also object identifiers (OID) can be analyzed. DV certificates only have domain name specified in subject whereas the OV also has Organization details whereas the EV along with the above**

				thing includes the name of website owner while clicking the closed padlock.
Subject Alternative Name(s) (SAN/UCC), if any	**DNS Name=*.www.tiktok.com DNS Name=www.tiktok.com**	**NA**	**NA**	**It's an organized method that makes it possible to associate different values with the security certificate.**
Certificate category: Single domain, wildcard, Multi-domain SAN/UCC cert?	**Wildcard cert**	**Single Domain cert**	**Single Domain cert**	**Specifies whether the certificate applies to single domain, all subdomain under single domain or discrete domains**
Public Key Info like key algo, key length, public exponent (e) in case of RSA	**Algorithm: ECC Key length: 256 bits**	**Algorithm: ECC Key length: 256 bits**	**Algorithm: RSA Key Length: 2048 bits Exponent (17 bits): 01 00 01**	**The underlying algorithm that the client used for generation of public key along with its attributes**
Public key or modulus (n) in case of RSA	**04 17 33 c2 39 06 0b c8 f7 e7 3a 54 5a d2 b2 dc cb 4b 33 6e 71 45 65 85 e9 e2 47 5b 6c e3 38 6f 4f ea b7 74 0a 4a 3c 33 27 8c 5a 2d 21 95 89 60 55 e4 ea 5b 7d c3 96 6d 83 af ff d2 85 1e 42 c4 24**	**04 f2 f4 a4 15 25 f1 30 c6 73 01 1d 47 df 41 51 98 dc dc 73 bf cd cf d5 cf c8 9b 2a dc 00 a7 63 8c dc 85 0c 11 01 36 34 ab 5e 08 99 f4 a9 15 75 62 23 d4 7f ad d2 5a fd ca 31 94 05 14 67 e0 86 83**	**E2 3B E1 11 72 DE A8 A4 D3 A3 57 AA 50 A2 8F 0B 77 90 C9 A2 A5 EE 12 CE 96 5B 01 09 20 CC 01 93 A7 4E 30 B7 53 F7 43 C4 69 00 57 9D E2 8D 22 DD 87 06 40 00 81 09 CE CE 1B 83 BF DF CD 3B 71 46 E2**	**The public key that each of them have actually generated.**

			D6 66 C7 05 B3 76 27 16 8F 7B 9E 1E 95 7D EE B7 48 A3 08 DA D6 AF 7A 0C 39 06 65 7F 4A 5D 1F BC 17 F8 AB BE EE 28 D7 74 7F 7A 78 99 59 85 68 6E 5C 23 32 4B BF 4E C0 E8 5A 6D E3 70 BF 77 10 BF FC 01 F6 85 D9 A8 44 10 58 32 A9 75 18 D5 D1 A2 BE 47 E2 27 6A F4 9A 33 F8 49 08 60 8B D4 5F B4 3A 84 BF A1 AA 4A 4C 7D 3E CF 4F 5F 6C 76 5E A0 4B 37 91 9E DC 22 E6 6D CE 14 1A 8E 6A CB FE CD B3 14 64 17 C7 5B 29 9E 32 BF F2 EE FA D3 0B 42 D4 AB B7 41 32 DA 0C D4 EF F8 81 D5 BB 8D 58 3F B5 1B E8 49 28 A2 70 DA 31 04 DD F7 B2 16 F2 4C 0A 4E 07 A8 ED 4A 3D 5E B5 7F A3 90 C3 AF 27	
Key usages;	**Digital Signature,**	**Digital**	**Digital**	**The difference in**

how do they vary in the chain, mention in the remarks?	**Key Agreement (88)**	**Signature, Certificate Signing, Off-line CRL Signing, CRL Signing (86)**	**Signature, Certificate Signing, Off-line CRL Signing, CRL Signing (86)**	**key usage of end entity and that of Intermediate and root CA are in Certificate signing, off-line CRL signing and CRL signing. Which is intuitively as such privileges solely lies to CA and not the user**
Basic constraints, how do they vary in the chain?	**Subject Type=End Entity Path Length Constraint=None**	**Subject Type=CA Path Length Constraint=0**	**Subject Type=CA Maximum number of intermediate CAs: unlimited**	**It indicates the maximum depth of permissible certificate pathways that include the certificate and whether the present entity is a CA or not..**
Name constraints (if any), how are these useful?	**NA**	**NA**	**NA**	**It makes clear to the parties relying on it which namespaces are appropriate for different hierarchical names.**
Size of the certificate	**1,690 bytes**	**1,434 bytes**	**1,360 bytes**	**It's the size of the individual certificates that I have downloaded**
URI of CRL	**URI: http://cdp.rapidssl.com/RapidSSLECCCA2018.crl**	**URI: http://crl3.digicert.com/DigiCertGlobalRootCA.crl**	**NA**	**CRL URI specifies the specific location where a list of certificate that are revoked are listed, signed with the secret key along with**

				timestamp
URI of OCSP Responder	**URI: http://status.rapidssl.com**	**URI: http://ocsp.digicert.com**	**NA**	**OCSP is a server the the CA maintains which records the current status of online digital certificate**
Any other parameters that you found interesting? **Signed Certificate Timestamp List**	**04 82 01 6A 01 68 00 76 00 EE CD D0 64 D5 DB 1A** **CE C5 5C B7 9D B4 CD 13 A2 32 87 46 7C BC EC DE** **C3 51 48 59 46 71 1F B5 9B 00 00 01 8B B2 E9 08** **6B 00 00 04 03 00 47 30 45 02 21 00 AC F6 D8 60** **0D D4 79 6B 71 60 C7 9E 6C 69 95 2A C8 5A A1 06** **9F 4D 0C 6D 5E 3D 14 B2 24 B0 9D C6 02 20 2E E6** **C9 D4 1A 2E 40 C6 7E 2A 04 63 23 4B AD FA A6 91** **15 64 9D F5 6B E0 F9 0D 1A E4 12 4A C2 EB 00 76** **00 48 B0 E3 6B DA A6 47 34 0F E5 6A 02 FA 9D 30** **EB 1C 52 01 CB 56 DD 2C 81 D9 BB BF AB 39 D8 84** **73 00 00 01 8B B2 E9 08 83 00 00 04 03 00 47 30** **45 02 20 7A 4F D9 0D 45 B8 1E D8 CE 49 0D 17 D7** **92 E0 03 69 A5 F8 C3 D8 14 59 22 6B 8B 57 75 F8** **01 92 67 02 21 00 A0 0D 67 11 3A 27 F1 A1 C6 A8** **97 5E A5 91 E6 C7 D9 A3 0B 6D 78 0B 0C 5F FD D4** **A8 3F 9E CA 16 CE 00 76 00 3F 17 4B 4F D7 22 47** **58 94 1D 65 1C 84 BE**	**NA**	**NA**	**It indicates when the issuance occurred. The client is expected to confirm that the certificate in the log matches the one you are validating in order to validate the accuracy of the information.**

	0D 12 ED 90 37 7F 1F 85 6A EB C1 BF 28 85 EC F8 64 6E 00 00 01 8B B2 E9 08 B6 00 00 04 03 00 47 30 45 02 21 00 80 2B 23 96 83 74 E7 3D 89 AB DC F2 3E 59 F8 CA 5E AF 74 D0 AA 4B A6 01 67 B0 DB 4A BB A1 09 44 02 20 73 D4 67 C5 36 F6 8E 34 8A 1F 6B 64 D3 0A 71 FE 38 1C 53 AC B9 99 C9 D9 CB D5 9D 5A 25 77 76 5D			

Answer the following queries after filling out the above table:

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

Ans: Anyday EV is more trustable, secure and expensive then the others. Since EV stands for a greater level of confidence than other standards, it must pass stricter validation procedures to ensure that the organization satisfies CA/Browser Forum requirements and is correctly registered with the relevant authorities in its country.

2. What is the role of the Subject Alternative Name (SAN) field in X.509 certificates?

Ans: Subject Alternative Name (SAN) / Unified Communication certificate (UCC) is an extension to X.509 certificates that makes it possible to associate different values with the security certificate.. Name includes email address, DNS name, URIs, IP address, Directory Names etc

3. Why are key usages and basic constraints different for root, intermediate and end certificates? What could go wrong if all of them have the same values?

Ans: The key usages define what different things the keys can be used for andIt indicates the maximum depth of permissible certificate pathways that include the certificate and whether the present entity is a CA or not. They should differ for root, intermediate and end certificates as we can't give the same level of authority to each and every entity. Also not every entity is root CA and also we should restrict the size of the chain.

The primary goal of the CA hierarchy model is trust and if we give the same values of key usage and basic constraint then there is no trust and it's equivalent to not having a CA's. Like anyone can impersonate anyone and the whole model of security would be disturbed.

4. What is the difference between Signature value and Thumbprint aka Fingerprint of a digital certificate?

Ans: Signature Value is generated by applying hash function to the to_be_signed component of the certificate and then encrypting its private key. It is used to guarantee the validity and integrity of the certificate, enabling users to confirm the signature and be assured that the certificate is legitimate and has not been altered by a reputable authority.

Whereas in Thumbprint is just a hash of the certificate including the signature value. It's a unique and combat representation of the certificate allowing for quick reference and human-readable verification.

5. Why do RSA key lengths increase over the years? Why is ECDSA being preferred over RSA now-a-days?

Ans: There is increase in the length of RSA Keys as it becomes tiresome for the attacker to crack the same, as it would be much computationally difficult to factorize the public key component n into two large prime numbers.

Elliptic Curve Digital Signature Algorithm (ECDSA) is increasingly preferred over RSA due to its efficiency and security benefits. ECDSA provides equivalent security with shorter key lengths, resulting in faster cryptographic operations. This efficiency is particularly advantageous in resource-constrained environments and reduces bandwidth usage, making it suitable for modern applications.

6. What are pros and cons of pre-loading root and intermediate certificates in the root stores of browsers and OSes?

Ans:

Pros:

1) The overload of verifying the trust chain of certificates would be reduced.
2) Any issue in case of missing intermediate certificates would be removed
3) The OS or Browser can constrain some security attributes of the certificate
4) Reduced dependencies on the online verification of certificate thus reducing the opportunity window for the attack

Cons:

1) As the number of Intermediate CA's are in denomination of a few thousands it would increase the memory overhead in storing all would be high.
2) Updates of the certificates would not be that dynamic leading to delay in recognizing newly issued certificates or revoking compromised ones
3) Irrespective of whether the end entity want a particular certificate, it will be forced to store all the Intermediate CA's certificate

4) A change in single intermediate CA's certificate would lead to change in all the devices root store.

7. Why are root CAs kept offline? How do they issue certificates to intermediate CAs?

Ans: The definition of trust relies on a trusted entity signing for an entity so to break the definition and to have a starting point of trust we keep root CA's offline. CAs are kept offline to enhance security, minimizing the risk of compromise and protecting the critical cryptographic material This isolation shields them from online threats, ensuring the long-term integrity and trustworthiness of the root key

When the intermediate CA's signing request goes to the Root CA which proposes some set challenge to the intermediate CA's depending upon what kind of certificate type the other is demanding. Once these challenges are completed it is the root CA responsibility to verify that the challenges are satisfied or not. And then the certificate issuance by the root CA happens by signing the CSR with its private key.

8. Why are root and intermediate certificates of new CAs cross-signed by the legacy CAs? Answer this by taking Let's Encrypt and its parent organization as root and intermediate CAs.

Ans: Let's Encrypt initially cross-signed its root and intermediate certificates by an existing CA, IdenTrust, to ensure immediate trust in existing systems that might not have the Let's Encrypt root certificate in their trust stores. This cross-signing allowed Let's Encrypt to rapidly establish trust without waiting for its root to be widely recognized. The cross-signed certificate acts as an interim solution to bridge the gap between Let's Encrypt's establishment and broader recognition of its root certificate.

9. What challenge is posed to the certificate seekers (Alice) by Let's Encrypt CA before issuing wildcard certificates? How does Alice respond to it so that she passes it?

Ans: Let's Encrypt imposes a DNS-01 challenge. In the DNS-01 challenge, Alice must add a specific DNS TXT (text) record to the DNS records of the domain for which she wants the wildcard certificate. The TXT record contains a token provided by Let's Encrypt as part of the challenge. This proves to Let's Encrypt that Alice has control over the domain's DNS configuration. By successfully completing the DNS-01 challenge, Alice proves to Let's Encrypt that she has the necessary authority over the domain and can be issued a wildcard certificate for it.

10. List out names of OS/Browser/Company whose root stores were pre-populated with Root and Intermediate CA certificates of the website www.tiktok.com?

Ans: There is no central repository providing and maintaining which OS/Browser/Company pre-stores whose certificates. But talking about digicert, its certificate is almost stored in every

famous OS/Browser/Company. Being specific to my devices and the browsers that I used, digicert was present in windows 11 os root store and brave, chrome, microsoft edge browser. My intermediate CA rapidssl was not present in operating system root store but was present in all above mentioned browser's root store i.e. in microsoft edge, chrome and brave

PART-B

1. A browser X has received the digital certificate of the website #N over a TLS connection. How does it verify whether the certificate is valid? Write a psuedo-code of browser X's verifier function named myCertVerifier( ) and explain how it works by picking the entire chain of trust of an end-user cert (of the website #N) in PART-A of this assignment.

Ans:

```
function myCertVerifier (CertificateChain CrtChain){

        While (CrtChain != NULL)
        {
                Crt = CrtChain.certificate
                IsValid = checkValid(Crt)
                notExpired = checkExpired(Crt)
                isSameDoamin = checkDomain(Crt)
                isNotRevoked = checkCRL(Crt)
                If ( !IsValid or  !not Expired or !isSameDomain or !isNotRevoked){
                        Return invalid
                }
                CrtChain = CrtChain.previous
        }

        Return valid

}

function checkValid(Certificate Crt){

        hash = hash_algorithm(Crt.to_be_signed)
        decrypted_hash = signature_decryption_algorithm(Crt.signature_value,ca_public_key)
        if( hash == decrypted_hash){
                return success;
        }
        return NULL

}
```

```
function checkExpired(Certificate Crt){

        if( Crt.valid_after <= os.today() <= Crt.valid_before )
                return success
        else
                return NULL

}

function checkDomain(Certificate Crt){

        If input_domain == Crt.CN // checking whether the certificate matches the expected
                Return success;    // domain name
        Else
                Return NULL

}

function checkCRL(Certificate Crl){

        Isrevoked = fetchCRL(Crt.CRL_URL) // or using OCSP or OCSP stamping
        if(Isrevoked)
                return NULL
        else
                return success;
}
```

2. Consider the scenario in which evil Trudy has used the domain validated (DV) digital certificate of the website (Bob) named Bob.com to launch her own web server with the domain name, **xyz.com**. Does your function myCertVerify( ) returns valid or invalid for this when someone like Alice (browser) tries to access Trudy's website xyz.com?

Ans: myCertVerify() will return invalid for the above mentioned scenario as in the checkDomain() function of the above mentioned pseudocode we are validating the authenticity of the domain name typed in the URL with the received certificate subject/Common name. Thus ensuring that what is received is the intended certificate of the domain we are requesting.

3. Consider another scenario in which evil Trudy has used the digital certificate of Bob's website Bob.com to launch her own web server with the domain name, **xyz.com**. When a web client (Alice) tries to connect with Bob's website by sending a DNS query, Trudy responds with her IP address by DNS cache poisoning (What is DNS cache poisoning? |

DNS spoofing | Cloudflare) Does your function myCertVerifier( ) returns valid or invalid for this and what are the consequences? What kind of attacks can Trudy launch in this scenario?

Ans: DNS cache poisoning or DNS caching just redirects the bob.com request to xyz.com by replacing the IP address of bob.com with xyz.com in the DNS resolver cache. But as mentioned in the above pseudocode the ***checkDomain()*** checks whether the certificate matches the expected domain name or not, and that will easily catch the falsy trudy website xyz.com.

- In your report, briefly explain how 7-zip uses the password to encrypt compressed files using secure hash and symmetric algorithms. What role does the password length play in brute force attacks to decrypt the encrypted files?

- **Ans:**
- **7-zip** uses PBKDF to generate the required security parameters that are fed to the underlying encryption algorithm. This PBKDF inturn uses slow hash like bcrypt, ecrypt etc over iterations of some thousands and then generates the required things. When a user enter the password that should protect the file, it picks a random salt and then give both as input to the slow hash function which produces the required security parameter (IV and key in case of AES) which are then given as input to the underlying encryption algorithm using which the encryption of the required files take place.
- Password length plays an important role to shield against any brute force attacks. Increasing the length of the password makes it tiresome for the attacker to crack the decryption in case of data breach. When a data breach occurs it generally leaks the salt and the IV of that file, then the attacker has to rely on a password generator to randomly find a password that matches the leaked configuration to crack the file. Increasing the length of password would increase the computational constraint exponentially and would also increase the time to crack the same.

References:

1. **https://crt.sh/**
2. **https://ahrefs.com/blog/most-visited-websites/**
3. **http://lapo.it/asn1js/#**
4. **http://phpseclib.sourceforge.net/x509/decoder.php**
5. **https://www.ssl.com/article/dv-ov-and-ev-certificates/**
6. **https://www.ccadb.org/**
7. **DV, OV, IV, and EV Certificates - SSL.com**
8. **7-Zip (7-zip.org)**

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 material 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 plagiarized 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 it.

Name: Yug Patel
Date: 06/02/2024
Signature: Yug Patel