Problem set 2, Brennan Giles

1. Private key is faster than public key, which takes extensive computations and time to complete. Private key is symmetrical and there is only one key (the other is a copy) while with public key it is asymmetrical and there are two. Lastly, with private key the two parties have to meet beforehand in order to share the key or it doesn't work, whereas with public key it could be two strangers that live in different countries and they can still communicate securely.

2. A digital signature is a process that guarantees that the contents of a message have not been altered in transit. It provides the security property of authentication, integrity, and non-repudiation.

3. MAC is symmetric while digital signature is asymmetric. MACS don't have non-repudiation security property either.

Here is a cool diagram that I'm including mostly so I can use it to study later:

```
Cryptographic primitive | Hash |    MAC    | Digital
Security Goal           |      |           | signature
------------------------+------+-----------+-------------
Integrity               |  Yes |    Yes    |   Yes
Authentication          |  No  |    Yes    |   Yes
Non-repudiation         |  No  |    No     |   Yes
------------------------+------+-----------+-------------
Kind of keys            | none | symmetric | asymmetric
                        |      |    keys   |    keys
```

4. A) She needs to send message m encrypted by bobs public key PKb.
   b)He should not send his CC number; since the message and key are enciphered with Bob's public key, he doesn't know it is Alice sending the message and not a malicious third party. The message should have been sent with a Digital signature.
   c)She should just tell Bob in person. Assuming the question wants an answer about encipherment, use symmetric key crypto.

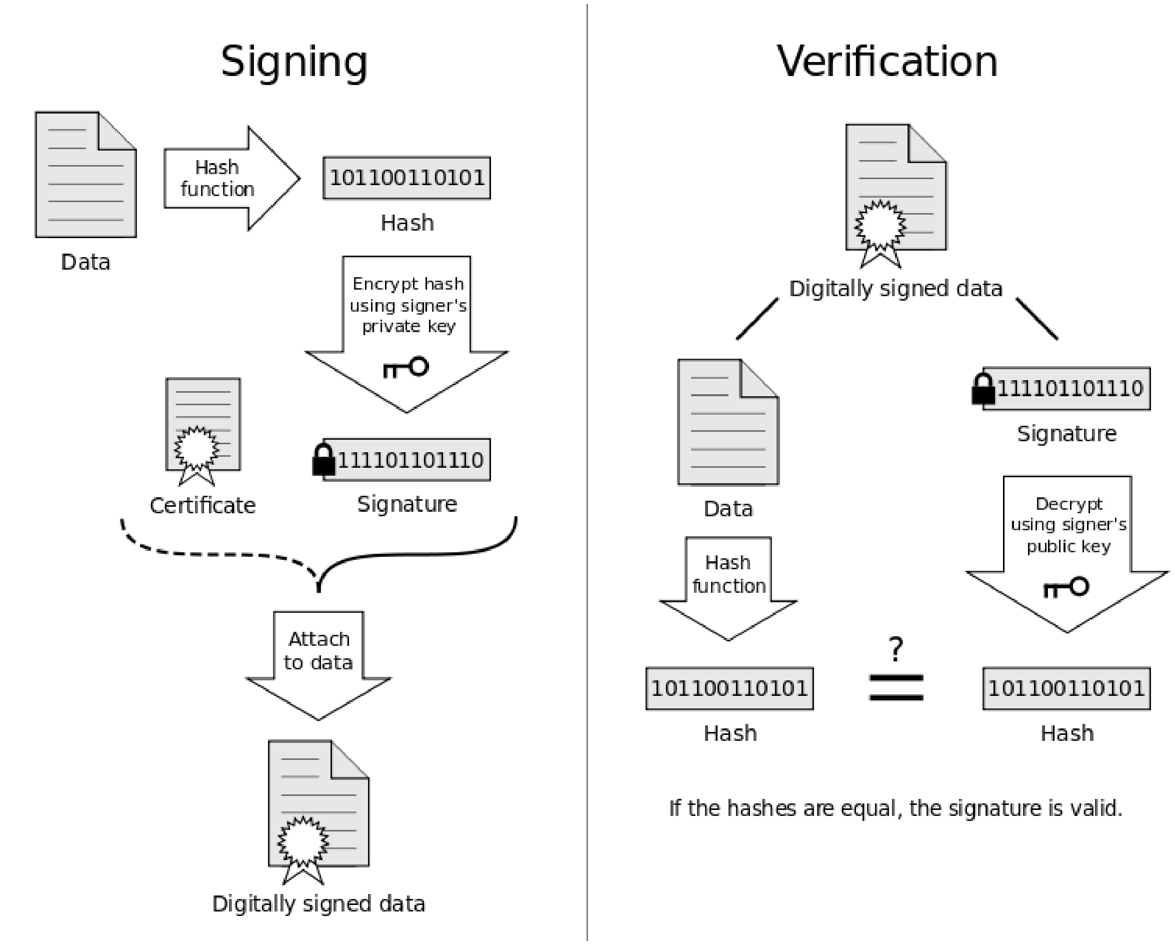
5. Yes they do, because they are based on the original message plus the signature.

6. m^d mod n

7. Man in the middle is an interception attack where a third party intercepts a message and sends their own to the intended recipient, pretending to be the person who sent the intercepted message. Meet in the middle doesn't allow for altering or sending their own messages posing as the sender, and is intended instead to discover encipher and decipher keys.

8. yes Because without it allows for forgery and the authentication property is lost. The hash before the signing prevents the adversary from reordering and existential forgery.

This image is for my own study purposes (digital signature)



9. no because just with the hash function it is enough to guarantee security properties

10. sign then encrypt, that way signing information is protected by the encryption. It is only until it is decrypted by private key that the verification can be accessed which is good for safeguarding against authentication attacks.

11. a digital certificate provides proof that the sender of a message is who they proclaim to be. Often used with digital signatures as seen above. A certificate chain is an ordered list of certificates containing SSL certs and CA certs that enable the receiver to verify that the sender and all CA's are trustworthy.

12. a)They are "Nonces" or pseudorandom numbers. They can only be used once!

b) performing a simple operation on the nonce confirms initial knowledge of the nonce that was sent by Bob, gives authentication to alice for bob's sake.

c) timestamps are always fresh and don't need to be secret. Also you don't need to keep a log for them so they are simply easier to use. So to answer, yes.

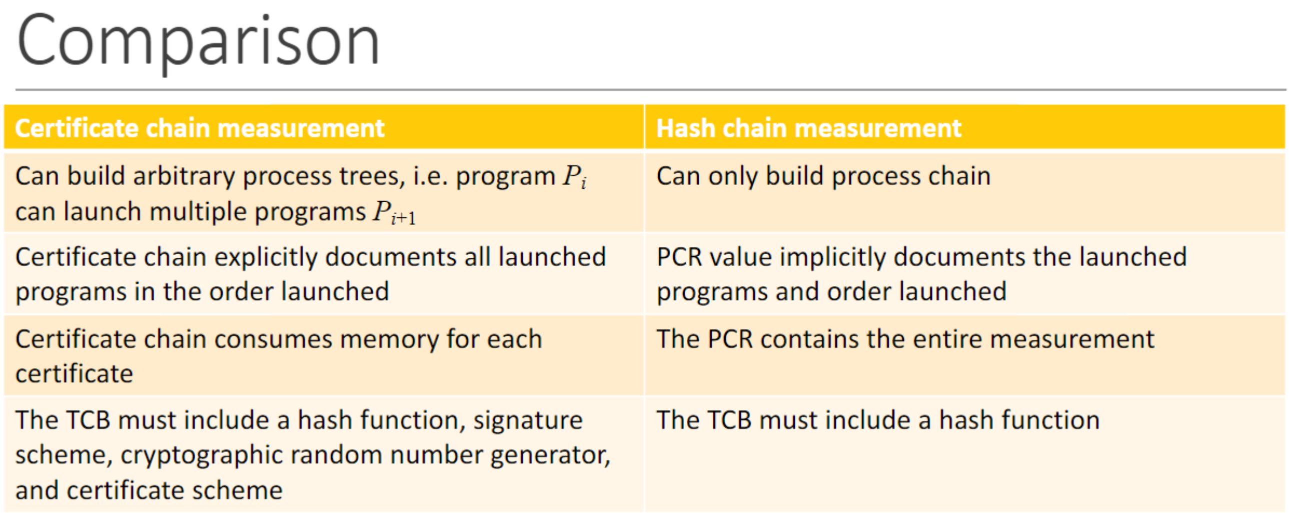
13. a) measurement problem: measuring what program loaded

attestation problem: reading our measurement the policy problem: how only authorized programs execute

the ownership problem: who gets to authorize

b) a hash function successively applied to layers of data in order to record the chronology of the data as well as ensure it's integrity.

“compare and contrast”:



Comparison

Certificate chain measurement	Hash chain measurement
Can build arbitrary process trees, i.e. program P_i can launch multiple programs P_{i+1}	Can only build process chain
Certificate chain explicitly documents all launched programs in the order launched	PCR value implicitly documents the launched programs and order launched
Certificate chain consumes memory for each certificate	The PCR contains the entire measurement
The TCB must include a hash function, signature scheme, cryptographic random number generator, and certificate scheme	The TCB must include a hash function

c) A manifest is a special type of certificate chain in which the subject is the digest of the program to which the manifest is loaded, and that the cert is signed by the program manufacturer. It asserts that the manifest subject identifies the software it is intended to release.

d) TCB is hardware based and can be accessed with special tools, so tcb would not help us trust that remote machine since the adversary could potentially compromise the device.

e) no it cannot, only what was loaded.

- f) I'm keeping this information the way I found it for study purposes:
- In a Secure Boot chain, each step in the process checks a cryptographic signature on the executable of the next step before it's launched. Thus, the BIOS will check a signature on the loader, and the loader will check signatures on all the kernel objects that it loads. The objects in the chain are usually signed by the software manufacturer, using private keys that match up with public keys already in the BIOS. If any of the software modules in the boot chain have been hacked, then the signatures won't match, and the device won't boot the image. Because the images must be signed by the manufacturer, it's generally impractical to sign any files generated by the platform user (such as config files).
- In a Measured Boot chain, we still depend on a Root of Trust as the starting point for a chain of trust. But in this case, prior to launching the next object, the currently-running object “measures” or computes the hash of, the next object(s) in the chain, and stores the hashes in a way that they can be securely retrieved later to find out what objects were encountered. Measured Boot doesn't make an implicit value judgement as to good or bad, and it doesn't

stop the platform from running, so Measured Boot can be much more liberal about what it checks. This can include all kinds of platform configuration information such as which was the boot device, what was in the loader config file, or anything else that might be of interest. Secure Boot is relatively self-contained. If the handful of signed objects haven't been tampered with, the platform boots, and secure boot is done. If objects have been changed so the signature is no longer valid, the platform doesn't boot and a re-installation is indicated.

https://forums.juniper.net/t5/Security/What-s-the-Difference-between-Secure-Boot-and-Measured-Boot/ba-p/281251

g) A method of protecting information by binding it to a software platform configuration. Basically, only a specific combination of software and hardware allows the data to be released.

h) taking ownership and ownership transfer. Only the taking ownership model is implemented because it takes more machinery and infrastructure to support the ownership transfer method in order to keep track of the history and signatures

14. a) since computers cannot create truly random numbers, we get as close to it as possible with pseudorandom numbers which emulate truly random values. They are used in crypto often, especially in order to choose salts seeds and keys. The more random the better because patterns make keys predictable and able to be compromised.

b) interchange key is used to identify who is who, and session keys are used to encipher messages and should be changed for each session. Interchange keys are changed independently of communication between the two parties. Session keys are needed to ensure the messages retain message integrity. Yes we still need session keys even when we have interchange keys because interchange keys only identify each person as the person they claim to be but does not ensure the message is protected.

c)because mutual authentication is simplified and made much easier when a third party organizes and handles all of it for the potentially thousands of people involved that need the authentication; true for both symmetric and asymmetric.

13. mutual authentication is a method of authenticating two parties while multi factor is merely authenticating a single person in multiple ways.

14. a) Authentication, Authorization, Audit

b) in DAC, regular users can adjust the policy, but not in MAC. In Mac each user gets to access according to their "level" of access, with DAC each resource has a list of users that is allowed to access it.

c) closed: access limited to those explicitly stated – default deny. Open: access limitations are specified, all others allowed.

d) RBAC is focused on roles to determine policies while ABACK is focused on attributes of user, resource, environment, or context to determine policy

15.

a) dev code/ exec: rights to r/w, delete, execute, promote

testing code/ exec: r/w, delete, execute, promote, demote

test reports: read, write

production code/ exec: read, execute

b)

	Product managers	Programmers	Testers	everyone
Dev code	read	read		
Dev code		write		
Dev code		delete		
Dev code	execute	execute		
Dev code		promote		
Testing code			read	
Testing code			write	
Testing code			delete	
Testing code			execute	
Testing code			Promote	
Testing code			Demote	
Test reports			Read	
Test reports			write	
Production code	read			read
Production code	execute			execute

c)

Product Managers: read, execute

Product Managers own Programmers

Programmers: read, write, delete, execute, promote

d)

Alice has access to Dev code for reading, writing, deleting, executing, and promoting

Can also read test reports, production code and execute production code.

16.

a) Not allowed. C does not own or have transfer rights for file 2.

b) Allowed, A owns C and thus inherits their rights on file 2.

17? Its not numbered

a) iii

b) ii,iii, V

c) I, ii, iii

d) I, iii