CISSP Domain 3 — Security Architecture and Engineering

Domain 3 — Security Architecture and Engineering

Security engineering and security architectural design are, of course, *far* more complicated and difficult than (ISC)^2 material suggests. See the work of Ross Anderson and Matthew Green for the real stories.

Ross Anderson

Matthew Green

Formal Security Models

These are formal and academic. They're listed here in order from most to least likely to appear on the exam. All you need to know about them is:

Bell—LaPadula model

The easiest and most obvious, as it's focused on secrecy. **Secrets should not flow down** in a multi-level security model.

Top Secret
Secret
Confidential

Someone with a *Secret* clearance should not be able to:

- Read *Top Secret* data
  (no *read up*)
- Write into a *Confidential* space
  (no *write down*)

That's easy, so they put further labels on things:

Simple Security property	No read up
Star property	No write down

Biba model

Conversely, focused on **integrity** and therefore **trustworthiness.** The easiest analogy is to think of a village with a monastery, where the monks are copying scriptures. You only trust the integrity, thus the meaning or accuracy, of a document from above: **Belief or trust flows down, not up.**

Chief priest
Monks
Villagers

- The chief priest, being the most enlightened being in the village, does not read or believe the possibly faulty copies of the monks, let alone the villagers' uninspired output.
- Monks read and believe what the chief priest gives them. They do *not* pay attention to pamphlets written by the villagers.
- Villagers should read and believe scriptures copied by the monks. They also should believe the less frequent publications from the chief priest.

This has similar formal properties with converse meanings:

Simple Integrity property	No read down (A subject cannot observe an object of lower integrity)
Star property	No write up (A subject cannot modify an object of higher integrity)
Invocation property	A subject cannot send logical service requests to an object of higher integrity. (The monks can't make requests of the chief priests, and the villagers can't ask anyone for help.)

For both Bell-LaPadula and Biba: *Reading is simple, all I have to do is look at the page. Writing is more involved, I must have a pen, and ink, and move the pen. So,* ***"simple" ones have to do with reading, the others, "star", with writing.***

Brewer and Nash, or the Chinese Wall model

This focuses on **preventing conflicts of interest.** A law firm or financial services firm might support clients who compete or conflict with each other. An individual staff member can choose to work with either client, but once they choose, they are limited to dealing with them only.

Wall Street

$9.99

Rated 4.6 out of 5 by 863 reviewers on Amazon.com

Buy Now

Wall Street: Money Never Sleeps

$14.99

Rated 4.3 out of 5 by 682 reviewers on Amazon.com

Buy Now

This is basically the movie *Wall Street*. Bud Fox could have worked for his father's union of aircraft mechanics, *or* for the Wall Street trader Gordon Gekko, but his penetration of the metaphorical wall leads to trouble for everyone.

Clark—Wilson model

Biba plus integrity at the transaction level, through the abstraction of well-formed transactions. **Think of Amazon.** They have an enormous and critical inventory database. Of course you aren't allowed to interact with it directly.

Shout-out to student John Bernheimer for providing the great analogy!

However, your interactions with their web pages do access the database through some limited, trusted, secure API. You click *"Put this in my cart"* and then *"Check out",* driving a defined sequence of transactions. Everything you do followed their defined rules, and the end result is that you *have* modified their database, through the process of purchasing an item. Not directly, all through their trusted API.

Graham—Denning model

Defines three categories: *objects, subjects,* and *rights.*

Then defines a set of eight primitive commands that subjects can execute to have effects on objects or other subjects.

It's a small set of basic operations, or verbs. Simple is easier to get right, so it's more secure.

Harrison, Ruzzo, Ullman model

Like Graham—Denning, but less restrictive.

Cryptography

I'm surprised that CISSP doesn't go deeper into cryptography than it does, given the exam's reputation as academic and rigorous. Here is a set of terms you need to know.

- **Plaintext** or **Cleartext**
- **Ciphertext** or **cryptogram**
- **Cryptosystem** is the entire system — the **cipher** or algorithm, plus all the details of how the keys are generated, agreed upon or exchanged, and used.
- **Cryptovariable** is obviously the key, plus maybe an IV or other data.
- **Initialization vector** or **IV** means that even if you encrypt the same cleartext with the same key, the ciphertext will be different. Patterns won't leak through.
- **Session key** means you use a new randomly generated key for each message, or for each encrypted file, or for each web or VPN session. Breaking one message gives your attacker just that, it does not give them anything else. (and see the cryptanalytic attacks below)
- **Key space** and **work factor**
- **Symmetric** versus **asymmetric**
- Asymmetric ciphers include
  - **Diffie-Hellman** (not really, but for the test...)
  - **El Gemal** (Professor El Gemal figured how to do what Diffie and Hellman were trying to do, when they instead discovered a secure way to agree on a shared secret in an insecure environment)
  - **Elliptic-curve** or **ECC**
  - **RSA**
- **Stream**

    - **RC4** (we shouldn't use it, but it's entrenched in the test just as it was in the real world)
    - **Salsa** and **ChaCha,** used in TLS
- **Block**
    - **DES** and **3DES** because they have been so entrenched
    - **AES** aka **Rijndael**
    - **Blowfish**
    - **Twofish**
    - And modes
        - **Electronic Code Book** or **ECB**
        - **Cipher Block Chaining** or **CBC**
        - **Cipher Feedback Mode** or **CFM** (makes a block cipher stream-like)
        - **Output Feedback Mode** or **OFM** (makes a block cipher stream-like)
        - **Counter Mode** or **CTR** (makes a block cipher stream-like)
        - **Counter Mode with CBC Message Authentication Mode Protocol** or **CCMP** (part of 802.11i or Wi-Fi security, stream-like *and* authenticated)
- **Encoding** and **decoding,** which change format without providing security. For example, HTML representation of Unicode, like `&#x0429;` for Cyrillic Щ instead of the UTF-8 character. The exam does *not*consider the use of code words for security, such as the Japanese Imperial Fleet first replacing `MIDWAY` with `AF` before encrypting that with the JN25 cipher.
- **Hybrid cryptography** appears on the exam as: My message to you starts with a header: *"Let's use AES with randomly-generated key* `01101000101...`*",* encrypted with an asymmetric cipher using your public key. Then the bulk of the message is the result of what the header said: using a symmetric cipher (like AES) and that key. Symmetric is computationally efficient, but we need a shared secret key. Asymmetric lets us share the secret.
- **Substitution** versus **transposition**
- **Monoalphabetic substitution cipher,** big name for a trivially broken cipher, see Poe's story "The Gold Bug" for a tutorial on how to do that. A **Polyalphabetic substitution cipher** was also broken back in Poe's time, it's a little more work but still easily broken with pencil and paper.
- **One-Time Pad** is unbreakable *if* correctly used, but impractical for most situations.
- **Steganography** and **covert channel** and **null cipher,** different but related.
- **Confusion** versus **Diffusion,** and, related to diffusion, especially for a stream cipher, **functional complexity**so each keystream bit depends on most or all of the cryptovariable bits
- **Kerckhoff's Principle** — if you think you have to keep the algorithm secret, then you're hiding a weakness. The strongest ciphers are those for which you don't worry about your adversary having a copy of the code. Or, really, having the entire **cryptosystem** *except* for the **cryptovariable** you used to convert your **cleartext**into **ciphertext.** See what I did there? *Make sure you can use these terms in sentences, because the exam certainly does!*
- **Avalanche effect**
- **Perfect Forward Security** or **PFS**
- **Collision** (issue with hash functions: multiple inputs produce the same output) versus **key clustering** (issue with ciphers, multiple keys produce the same ciphertext from the same cleartext)
- **Hash functions**
    - **MD5** because history
    - **SHA-1** should be history soon
    - **SHA-2** is a family including SHA-256 and SHA-512. May be written "SHA-2" to mean one of them, "SHA-256" and "SHA-512", or "SHA-2-256" and "SHA-2-512". Less likely, SHA-224 and SHA-384 could show up.
    - **SHA-3** — there was a big scare when the weaknesses in MD5 and SHA-1 came out, and US NIST announced a contest to replace the SHA-2 family. Keccak, designed by a team including a member of the Rijndael/AES team (Go, Belgium!) won, but... It turns out we didn't need SHA-3 immediately after all. Some day it will be a drop-in replacement for the SHA-2 suite.
- Cryptanalytic attacks
    - The situation:
        - **Ciphertext-only**
        - **Known plaintext**
        - **Chosen plaintext**
        - **Chosen ciphertext**
    - The approach:
        - **Linear cryptanalysis**
        - **Differential cryptanalysis**
        - **Side channel attack**
        - **Fault analysis**
        - **Probing attack**
- **Message Integrity Control** or **MIC** (it hasn't been changed, just a hash will do this) versus **Message Authentication Code** or **MAC** (it hasn't been changed *and* it came from a specific sender, so maybe a hash of the payload plus a shared secret key, thus an **HMAC**)
- **Link encryption** would be handled for you by the people running the link (satellite, fiber, etc). It encrypts the raw bit stream, headers and all, so the routing data (practically speaking today, the IP header) is also encrypted.
- PKI or Public Key Infrastructure
    - **Non-repudiation**
    - **X.509v3,** the standard format for a digital certificate
    - **Certificate Authority** versus a **Registration Authority**
    - **Certificate Revocation List** or **CRL,** which you might check via **OCSP**

Quantum Defense and Offense

Quantum cryptography is for *defense,* to *protect* secrets. It's used for **QKD** or **quantum key distribution,** where you use single-photon signalling to securely exchange a key to be used with a conventional symmetric cipher like AES.

Quantum computing is for *offense* or **cryptanalysis,** to *attack* secrets. A general-purpose quantum could quickly solve problems which are otherwise impractically difficult, thus breaking all the asymmetric ciphers we now use to protect key exchange: factoring (RSA), discrete logarithm (ECC, Diffie-Hellman, El Gemal).

Now That You Know Cryptography...

There are some questions where knowing all the technology doesn't give you the correct answer. You must carefully analyze the English prose.

$(ISC)^2$ isn't as bad as CompTIA about doing this, but they still do it on some questions.

Here's are two examples, from my CompTIA Security+ study suggestion page. In the second one, "BPA" means "Business Partnership Agreement". You can click "See the answer" to be taken to the answer and its explanation, and then "Return to main page" to get back here:

Example Question #1

Question: You want to use a system that can protect communication by authenticating the server, and also providing a copy of the server's public key in a trustworthy format. A provider of trusted certificates will only provide one when you follow their rules. There is a protocol that you can use to check in real time whether a certificate should be trusted or not. You must have a copy of the currently untrusted certificates locally, to reduce network traffic. Rather than a complete copy of the key, you may refer to its hash instead. There are ways to prevent a breach today from exposing secrets based on keys in the past. What do you need?

A: TLS
B: CPS
C: OCSP
D: CRL
E: thumbprint
F: PFS

See the answer

Example Question #2

Question: Your CEO has met with the CEO of another company, and they have agreed to work together to develop a new service. Authentication and identity management will be connected across the two organizations. Given the sensitivity of the development project, User authentication and authorization will use a centralized server running the best available trusted third-party service. Users will receive identity and service tokens from a unified authentication and authorization service, which requires that system clocks be synchronized across the organizations. Applications will be limited to those written with the API of that service. What do you need?

A: BPA
B: Federation
C: Kerberos
D: KDC
E: NTP
F: Kerberization