The World of Cryptographic Failures

This article will discuss Cryptographic Failures: what it is, how it happens, and why it’s dangerous.

What are Cryptographic Failures?

A cryptographic failure is a failure of cryptography or the implementation or use of cryptography. In most cases, it’s the implementation or use where the issue lies. Modern cryptography is very good, but just as the best lock in the world won’t be secure if it’s mounted on a door made of balsa wood, even the most secure cryptography won’t keep information secure if an attacker can bypass it entirely.

Why Do Cryptographic Failures Happen?

Reason 1: Bad Or Missing Cryptography

Not all cryptographic algorithms are equally strong. Some cryptographic algorithms that used to be considered strong are now weak, and many currently strong algorithms will likely be weak in a few decades. Cryptographic algorithms can become weak due to computer advancement or structural flaws in the algorithms themselves. Regardless, using weak cryptography is one source of cryptographic failures.

Another similar issue is not using any cryptography at all. This can include issues like forgetting to enable encryption for a database or disabling encryption for one reason or another, typically for compatibility. For example, some websites will fall back to using HTTP if HTTPS is unsupported, which is bad if you're going to be sending sensitive data to that website.

Reason 2: Key Generation And Storage Issues

Encryption algorithms rely on cryptographic keys to encrypt and decrypt information. However, a lot can go wrong when generating and managing these keys.

Firstly, when you generate these keys, you shouldn't use default keys or reuse keys. Using default keys is especially bad, but even reusing keys will reduce the overall security of the encrypted data. If a key gets stolen, you'd like it to open as little as possible, and using one key for many purposes goes against that.

Using passwords directly as keys is also a bad idea. There are ways to generate keys from passwords, but they shouldn't be used as keys. Generating keys from passwords is known as *password-based key derivation*, and there are right and wrong ways to do it.

Once the keys have been properly generated, they need to be *managed*. Key management technically includes key generation, but it also covers things like storage, distribution, and procedures for what to do when a key has expired or been compromised. Improper key management is another common cause of cryptographic failures.

Finally, don't write keys in source code or include them in source code repositories. This is a common enough issue that platforms like GitHub have implemented automated scanning tools to warn developers when a key has been included in a source repository. Ironically, GitHub itself once accidentally published one of its own private keys to a public repository in March 2023. Thankfully, the issue was detected quickly, and the key was promptly replaced.

Reason 3: Initialization & Randomness Issues

Not all randomness is created equal. In fact, many things that we refer to as random are completely deterministic - they only appear random because we don't have enough information to determine future results. A *random number generator* (RNG) is an algorithm that generates "random" numbers. RNGs themselves are deterministic and, if initialized to the same state, will produce the same outputs - more on that later.

Cryptography has stricter standards for randomness than other purposes, and using a bad RNG can be a serious security flaw. When a random number generator is good enough for cryptography, it is *cryptographically secure*. It is *very* important to use cryptographically secure random number generators. Insecure random number generators have patterns in their outputs that can allow an attacker to make predictions about what future outputs will be, which is bad news if those outputs are used to, for example, generate encryption keys.

RNGs need to be provided with an initial state using an *initialization vector* (IV), and it is very important that these initialization vectors be unpredictable. If you give two copies of the same RNG the same IV, they will produce the same series of outputs. Just like you need a cryptographically secure RNG, you also need a cryptographically secure method of generating IVs.

Why Are Cryptographic Failures Dangerous?

Cryptography has various uses and protects confidentiality, integrity, and availability: the three fundamental pillars of security. When a cryptographic failure occurs, security has been compromised in some way. That might mean that confidential data was exposed, that information could be modified without detection, that people couldn't access the resources they needed, or any other issues. With how much we use cryptography in our day-to-day lives, there can be any number of serious consequences resulting from cryptographic failures.