Full Chain Cryptographic Security

Full chain cryptographic security is when every function across an application responsible for returning privileged information is only able to perform this task by obtaining from its caller the *cryptographic ability* to do so. A *cryptographic ability* is distinct from a *software ability* such as setting permission bits or a standard password authentication in that software is able to make a mistake and return privileged information which its designer had not intended it to do, this is simply not possible if it is built correctly upon cryptographic protocols. *You don't accidentally break AES256*

In order to return privileged data from a filesystem, the operating system usually checks a files permissions to see if the requesting user has access permissions to that file. However, through a mistake of logic, the software could return information which the user did not have the authority to access. A cryptographic ability is provided to a function by sending it something like a key, hash, or password(which derives a key), which is subsequently used to retrieve the required information. Note here that regardless of the logic in the called function, it would not even have the *ability* to retrieve the information no matter how many mistakes were in the code. It would just fail.

An unencrypted filesystem therefore, though permissioned, has only *software security*, but an encrypted filesystem, which is unable to return plaintext data without a decryption key provided by its caller, has *cryptographic security*.

Objection

But an encrypted filesystem usually stores a key in memory, rather than prompting a user for a password each time, technically, this means that it does not have *full chain cryptographic security*, as the caller does not provide a password every time.

> Not sure if this is always true or not. You can have a long running service, that technically does not need to ask for a key because it never terminates, but if that service receives instructions and then dispatches them to other processes, it still is not receiving that ability to perform it from its caller.