Cryptographic Hash Functions

A *cryptographic hash function* has the same properties as ordinary hash functions: it is easy to compute, takes an arbitrarily long input string (or file), and it produces a random-looking, fixed-length, output string. For example, the most common hash function used today, SHA2-256, outputs 256 bits (32 bytes), no matter the length of the input.

For example, the SHA2-256 hash of the previous paragraph is:

```
766c8486ceab57ef792b8c0ad9c97a4f375430410a327adeea5c260f6579683e
```

in hexadecimal.[1]

However, cryptographic hash functions have three additional properties with specific names:

Collision resistance: It is effectively impossible—a more precise phrase is "computationally infeasible"—to generate two different files whose hash value is the same.

Preimage resistance: Given a randomly chosen hash value, it is computationally infeasible to find an input string whose hash is that value.

Second preimage resistance: Given a file and hence its hash value, it is computationally infeasible to find a different file that will have the same hash value.

By "computationally infeasible", I mean that it is not possible to try enough different values, probably ever. Consider a 33-byte string. There are clearly more 33-byte strings than there are 32-byte strings; thus, there *must* be collisions: there have to be some pairs (or larger groups) of input strings that have the same 32-byte hash. However, you can't possibly try all possible 33-byte strings; there are far too many of them. The proper time unit for how long this calculation would take is measured in lifetimes of the solar system…

Why is this important? In means that, in essence, every file has a unique hash value, which is why a file's hash is sometimes called its "fingerprint". For all practical purposes, a file's hash uniquely identifies the file. Now: consider child pornography. Child porn images are contraband: it is illegal to possess them. Calculating the hash of, say, every uploaded file and comparing it to a list of hashes of known contraband images will, with extraordinarily high probability, only identify such illegal files, with far higher accuracy than the drug-sniffing dog in Illinois v. Caballes, 543 U.S. 405 (2005).

So: if this scheme only detects contraband, would a government mandate for Facebook or Gmail to perform such a check be constitutional? (Effectiveness is

---

[1] Hexadecimal is base 16; the digits are 0–9 and A–F. Programmers generally use hexadecimal to represent binary values for compactness, readability, and the ease of conversion: every four binary bits are represented as exactly one hexadecimal digit.

another question, since a change to the value of a single bit of the file would completely change its hash value.)

Now imagine that someone has stolen a digital copy of a highly sensitive U.S. intelligence document, and that the government wants Facebook and Google and other communications providers to search all files transmitted over their systems for the hash of the file. Would this be constitutional *without* a warrant? Would it be constitutional *with* a warrant? (This is a modified version of a hypothetical asked by Professor Jonathan Zittrain at a very entertaining Socratic dialogue here.)

Bonus question: There is a technique known as a *Bloom filter*[2] that permits extremely efficient search to see if a given value is in a set, e.g., a set of hashes of known contraband files. However, Bloom filters can produce *false positives*—they can claim that something is in the set when it isn't. You can adjust your Bloom filter to any probability you wish of a false positive, and thus make them very rare. Assuming that comparing a file's hash with a list of hashes of known contraband is permissible under the Fourth Amendment, is using a Bloom filter permissible, since it can have errors? How rare would such errors have to be to make such searches acceptable?

---

[2] If you want the technical details: B. H. Bloom, Space/Time Trade-Offs in Hash Coding with Allowable Errors, 13 Communications of ACM 422 (Jul. 1970).