FAQ Secure Untrusted Data Repository (SUNDR) (2004)

Q: SUNDR provides a means to verify whether or not files were tampered with. But does it hide the actual files themselves from a malicious server / do any sort of encryption on them?

No. SUNDR as is provides integrity, but not confidentiality. It could be extended with confidentiality by encrypting files, which a number of systems had done before SUNDR, and why the authors don't focus on that problem.

Q: How slow/fast is the computation of an i-handle in a fairly large file system?

Computing SHA-1s is quite fast (~10 cycles per byte). Furthermore, if you change 1 block, you just recompute the sha1-1 on that 1 block, and then sha all the blocks to get you back up to the root. The signing part is the expensive part. The sha overhead is definitely not neglible, however, and the paper proposes to avoid the cost of recomputing hash trees over several operations by allowing an i-handle to store the hash and small log of changes that have been made to the i-table. This is also speeds up the checking for clients that have checked a recent version; they can just apply the several operations in the log.

Q: If a fork won't be detected in SUNDR unless two users communicate, are there any instances where users may not ever communicate with each other, and also what does this "communication" look like in this system or between users in a file system generally?

With communication the authors mean any out-of-band communication between users (e.g., email, chat, etc.). For example, if user A tells user B please look at file "x", and B doesn't see "x", then they know they server forked the file system (or user A is lying to user B). Other than out-of-band communication, the authors also suggest using a time-stamp box. In bitcoin, there is yet another method to settle on a fork, which we will discuss next week.

Q: What are the current applications of SUNDR in products or internal systems?

As far as I know, none. But the ideas are powerful and you see them in other decentralized systems such as git, ledgers, etc. The one commercial system that I know off that is directly influenced by SUNDR is keybase (which was acquired by zoom).

Q: In what scenarios would you choose to store data on untrusted servers?

You might not store your data intentionally on an untrusted server, but your trusted server may become compromised. The attacker may change files on your server and now you have a problem, because recovering from such an attack is difficult. If this server stores the source of Debian Linux, as in the attack mentioned in the paper, then the problem is very serious. Similar attacks happen: Canonical was compromised in 2019.