How to make authenticating videos

From time to time, you are going to need to authenticate something: a promise, a contract, a piece of fiction, you name it. By “authenticate”, I mean that your authorship or endorsement of the item can be recognized by others. Typically, this is done by a signature that people can read and recognize, based on the assumption that it is difficult for people to forge someone else’s signature.

In the digital age, it is possible to “sign” an item by producing a digital signature matching the item, involving a secret key. The digital signature is typically a piece of random-looking text, which accompanies the signed item. People wishing to verify the signature can do so using the public key matching the secret key that was used for signing. The process to make a public key from the secret key (there are several kinds, RSA and DSA being the most common), is one-way. This means that it is easy, in terms of computing effort, to get the public key from the secret key, but it is very difficult to get the secret key from the public key. The secret key is never revealed, but the matching public key is made public so people can use it. Those verifying the signature perform a mathematical operation on it, involving the public key, which detects whether or not the matching secret key and the particular item being signed were used to make that particular signature. If the answer is affirmative, they conclude that the person possessing the secret key, and no one else, had to be involved in making that signature, thus confirming endorsement.

The problem with digital signing is that we still can’t know for sure whether or not a particular public key belongs to a particular person, only that it matches a secret key purportedly possessed by that person. It is entirely possible for someone else to pose as the individual whom everyone else believes to be signer, since the actual making of the public key is not witnessed. To get around this, we have created “digital certificates,” which attach the digital signature of the public key in question, made with a secret key that we implicitly trust as belonging to a trusted party who has witnessed the making of that public key. Many digital protocols use certificates made by “Certificate Authorities” (CA) that are trusted directly by browsers and other programs.

Sounds complicated? Of course it is, but computer programs manage to keep things straight most of the time. There have been reports of CAs not living up to the trust implicitly placed on them, but we keep accepting them regardless because the system allows us to carry out “secure” bank transactions and other sensitive stuff online. Now, when actual people are asked to do signing, verifying, and certifying for more mundane things such as email, they either refuse or fail to do it correctly. It’s just too involved, and the pitfalls are too many to keep track of them all.

In my own encryption program, PassLok, I’ve tried to address many issues having to do with the current unfriendliness of public-key cryptosystems. Among other things, PassLok

uses neither certificates nor CAs, so the problem of authentication goes back to the start. How can a user make sure that a Lock (PassLok's term for public key) purportedly belonging to someone else actually does? Even worse, how can anyone be sure that the PassLok code itself, which traveled over the Internet from a source server to his/her device, is the authentic code and not something that looks similar but where the guts have been replaced by something that appears to be secure, but is not so in reality?

And this is the point of this article. You can provide pretty strong evidence of the authenticity of your Lock (or anything else digital) by making a video. Let's see this with an example. I have made the PassLok Lock matching my secret Key, and now I want people to accept it as authentic so they start using it to lock messages intended for my eyes only. Here is my Lock:

PL15lok=WsH3zTgZn8V3hnIqjdbfPus+5YF5n+LBRPuH9USMMp8izPv+hsLoZKv
+jaCFMapJFfiA11Q9yJU1K1Wo0TbjXK/=PL15lok

In addition to the initial and ending PL15lok tags, it comprises 87 base64 characters, which is too long to read without making mistakes, especially over the phone. Therefore, the first thing I do is click the ID button in PassLok in order to get the SHA256 hash of the important part of the Lock. This is what I get:

4d0b-5224-c0bd-a2e0-2574-1cfe-3e3c-f630-a64b-cf6d-104d-3caf-4f1e-f970-bcb5-ee33

I am using the SHA256 function because it is available as part of the SJCL functions, on which PassLok is based. I could also do a SHA-1 or even an MD5 hash (not recommended, since it has been proven insecure) for which there are many utilities available, bot native and online. The resulting hash will always have the same length whether the original item is long or short, which makes it particularly convenient for big items. In addition, it is made only of numbers and the letters "a" through "f", which makes it much easier to read.

Now I print this on a piece of paper, along with other information I may want to add, such as my name, etc. Then I open youtube.com and log in with my account because I am about to upload a video. If you don't like YouTube, there are other video services out there that will do just fine.

Shooting is about to start, so I make sure there is a good light and the computer microphone picks up my voice. On YouTube, making a video of yourself is as easy as clicking on "upload" (after login), followed by "Record" (under "Webcam capture"), and then "Start".

But before I record myself reading this SHA256 string, I add a little twist. I pull out my smartphone and select a suitable background music, which I start playing next to my computer (so the microphone picks up the sound) right before I click on "Start". Why do I

do this? When I read the hash, I am going to be reading a sequence of only sixteen different characters. Someone who wants people to accept as authentic a counterfeit Lock could conceivably take the original video, cut it into pieces where I read just one character, and then reassemble it so I appear to be reading the hash of the counterfeit Lock. But if I have background music he can't do that without messing up the music, and people will realize something's wrong. It's like the security paper that agencies and schools use to issue certificates of any kind, only in sound.

And then I read the hash on camera, announcing first what it is that I am authenticating and, for better security, what the piece playing in the background is supposed to be. I also like to show to the camera the piece of paper that I'm reading from, so watchers can have yet another hard to fake way to get the hash. The whole recording takes about a minute.

As Youtube processes the recorded video, I add a copy of the hash to the description, so people can double-check, plus other interesting stuff such as the Lock itself, and then I copy the URL of the video. In the case of my PassLok Lock, this is the URL:

http://www.youtube.com/watch?v=5RzIpQhjMKE

And that's it! I can give this to people along with my Lock, or just this link if I have added the Lock itself to the video description. I can email it, text it, put it in a QR code, add it to my email signature or my calling card. The idea is to disseminate it widely.

A malicious third party who wants others to use a fake Lock under his/her control would have to:

1. Hack into Youtube to get access to my account: probably not so easy for a private hacker but a piece of cake for a government agency, given Google's previous history.

2. Change the description: trivial once they've gotten access to my account.

3. Load a counterfeit video. If they chop it up in order to make me say something different, they'll have to remove the background music first, which is hard without leaving artifacts, especially if the playback was mono rather than stereo and cheesy-sounding. Defective sound is harder to fake convincingly than quality sound. Alternatively, they'd get someone who looks and sounds like me and have him read the string. All of this amounts to getting into "Mission Impossible" territory, whether you are an individual or the NSA.

Of course, this only works for watchers who know what I look and sound like. It will work for friends, associates, and likely for strangers I've talked to recently. If I'm Justin Bieber, it will work for millions of teenage girls, too. So, gauge your celebrity level ahead of time and

use this technique judiciously. Who knows, you might be more popular than you ever thought you were ;-)