wavestore
integrity in digital recording

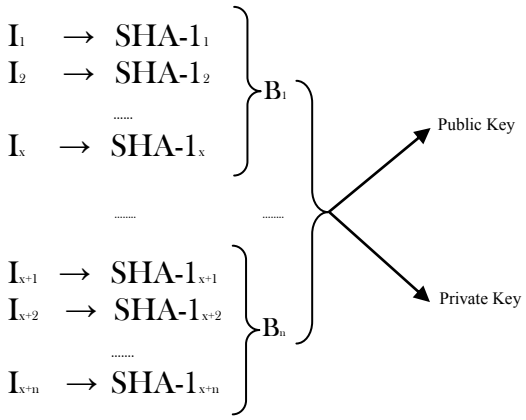
Watermarking

Wavestore Image Authentication

The aim of the Wavestore Image Authentication process is to be able to prove that an image or set of images were originally recorded on a particular Wavestore digital video recorder and that they have not been manipulated in any way. This mechanism also allows detection of images that have been modified after the initial recording. Authentication is only possible on recorded images, and there is about a 4 second delay before recorded images can be verified.

The core method for achieving image authentication is a digital signature . However, applying a digital signature to every recorded image would be very computationally expensive. For example, recording 800 images per second is quite achievable on Wavestore, but if each image were to be digitally signed, most modern CPUs would not be able to keep up. For this reason, a slightly more complex scheme is used to achieve the end goal.

For every image a SHA-1 hash is generated. Essentially SHA-1 is a cryptographic hash function which generates a unique code associated with the image data; the output data is much smaller than the image meaning that it can be digitally signed more efficiently.



To further improve the authentication performance, the image hashes are grouped together into 1-second blocks. It is this block of image hashes which is digitally signed using the RSA algorithm. For each Wavestore there is a single stream of authentication data (the blocks of hashes). This data is written at the same time as the video data, even though it is a separate recording stream. It is not interleaved with the video.

The mechanism of signing and checking the digital signatures requires the use of digital certificates provided by the PKI (Public Key Infrastructure) which is essentially based on a public and a private authentication key. In the Wavestore Image Authentication process, the private authentication key is stored the Wavestore server while the public authentication key, as the name suggests, is made publicly available to the user.

The private authentication key is installed on a particular Wavestore and is globally unique. The chances of an identical key existing in the world is very very slim. The public authentication key matches the private proving that the keys match, but it is virtually impossible to figure out the content of the private authentication key from the public one.

Therefore, the mechanism is that the blocks of image hashes are digitally signed using the unique private authentication key which exists only on that particular Wavestore. Users are given the matching public authentication key which is suitably labelled (i.e. "Acme Co. Wavestore 1 Public Key"). The user can attempt to match the public authentication key against the digital signatures on the block of image hashes. If the signatures are from a different device, the signature check will fail. Also if any of the images have been manipulated in any way, the signature check will fail. Thus we have achieved our original aim.

Certificate Generation

The certificate generation process can be described as follow:

- the Wavestore Client sends a request to the server requesting that the server generates a *"public/private authentication key pair"*;
- the server stores the *"private authentication key"* - this private key is never released;
- the server returns a digital certificate request containing the *"public authentication key"* to the client;
- the user sends the certificate request to Wavestore Limited who generate a certificate from the certificate request
- Wavestore Limited digitally signs the certificate using the *"Wavestore master private key"* to confirm that ownership by the user of that certificate (and therefore the contained *"public authentication key"*) is verified by Wavestore Limited - this can be checked later using the *"master public key"* which is embedded in the Wavestore client and Waveplayer playback software.

At this stage, the user has a digital certificate containing a *"public authentication key"* with verified ownership.
The user now loads the certificate onto the Wavestore so that it can be distributed with any copies of the recordings made by that Wavestore.

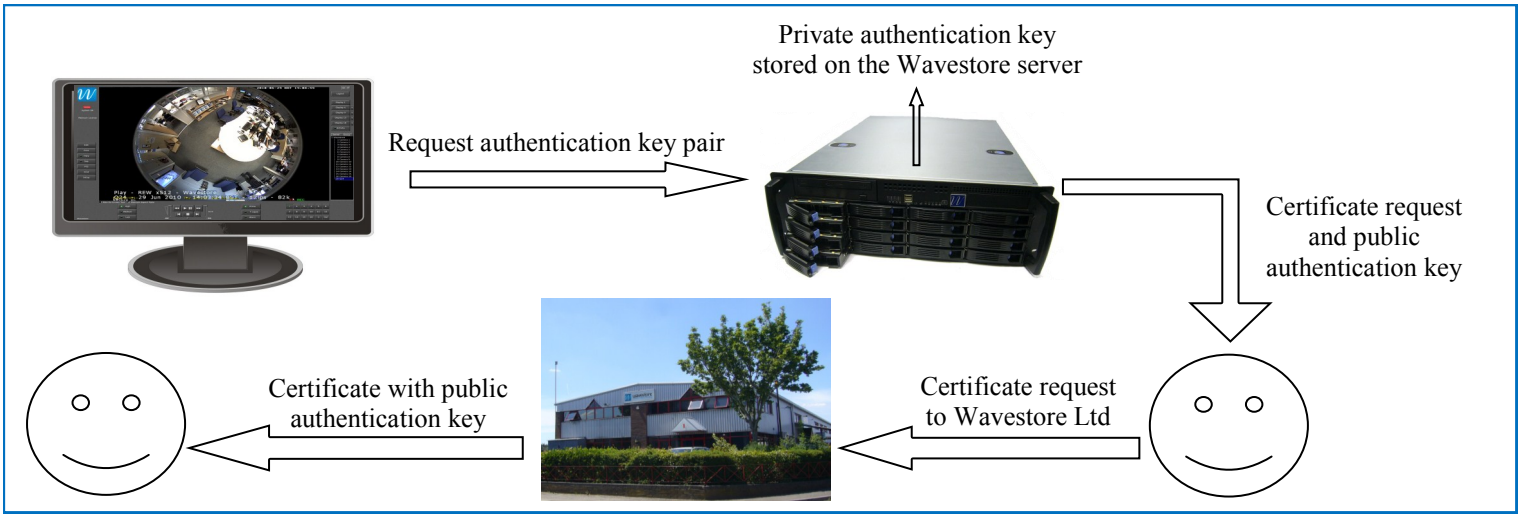


Image Verification

The user may want to verify either the authenticity of images on the Wavestore server itself, or images which have been copied from the Wavestore as part of a backup.
To allow backed up images to be verified, the backup should contain the stream of 1-second blocks of authentication data as well as the actual video streams. Inclusion of the authentication data is optional in the Wavestore client backup screen.

In order to verify an image:

- The 1 second block of image hashes is read - it contains the *"public authentication key"* certificate as well as the block of image hashes;
- The validity of the certificate is checked using the *"public master key"*;
- Assuming the certificate is proven valid, the *"public authentication key"* can be used to prove the validity of the block of image hashes;
- Assuming the image hash block is valid, meaning that it has not been tampered with, the hash corresponding to the original image is extracted - It is stored along with the camera details and timestamp so that it can be looked up; we'll call this hash the *"authentication hash"*;
- The original image is then read and its hash is generated. We'll call this the *original image hash*.

Finally, the *"original image hash"* is compared with the *"authentication hash"*. If they match, we know that the image has not been manipulated.