

Understanding and Integrating KODAK Picture Authentication Cameras

Introduction

Anyone familiar with imaging software such as ADOBE PHOTOSHOP can appreciate how easy it is to manipulate digital still images. Normally a compelling feature, the ability to modify images can be a serious problem when it enables business or legal records to be altered without detection.

Increasing digitization of documents has raised concerns about maintaining the integrity of digital data as well as detecting and deterring fraud. In response to these concerns, Kodak has developed a Picture Authentication Module for the DC280 Camera to test captured images for authenticity with a high degree of confidence.

While the concept of authentication is fairly straightforward, it is still new to most people. Like leaving your keys in the car or your house unlocked, it is possible to have a top-notch security system that isn't managed well. The key is to understand the basic principles.

Since it is relatively easy to design a workflow that incorporates picture authentication, it is likely that many businesses will choose to do so. The purpose of this document is to help you understand how KODAK Picture Authentication Cameras use encryption techniques to improve the security of your pictures. This document also explains how to integrate KODAK Authentication Cameras into your business workflow.

Basic Digital Signature Technology

Kodak has chosen to incorporate the Digital Signature Standard¹ (DSS) in its authentication product. The DSS technique is simple, robust, and sanctioned by the National Institute of Standards and Technology. The DSS uses public-key cryptography to carry out digital signature generation and verification. In the DSS scheme, the person or entity generating the digital signature (the signatory) holds a unique, secret key (the private key). An associated unique public key is freely distributed. The public key only allows verification of a digital signature created with the private key; it does not allow creation or modification of a digital signature.

Broadly speaking, authentication is the process of determining whether an item has been altered since it was "signed" (that it is the same as it was when it left the signer's possession). Using a digital signature is a relatively simple way to enable authentication.

Referring to figure 1, the process begins with a message that is to be transmitted. The first step is to represent the "state" of the message in a more condensed form by running the Secure Hash Algorithm (SHA-1) on the message, which generates a 160-bit message digest from the original message. The message digest is then input to the digital signature algorithm, which utilizes the signatory's private key to produce the digital signature. The digital signature is then appended to the message, and sent to the appropriate recipients. In essence, the digital signature is a "locked" record of the "state" of the message at the time it was signed.

When the message is received, it passes through the same Secure Hash Algorithm, producing a new 160-bit digest value. The new message digest and the original signature are input to the digital signature verification algorithm, along with the signatory's public key. The verification algorithm essentially uses the public key to unlock the digital signature and then verify it against the new message digest. If it verifies, the message has not been changed since it was signed.

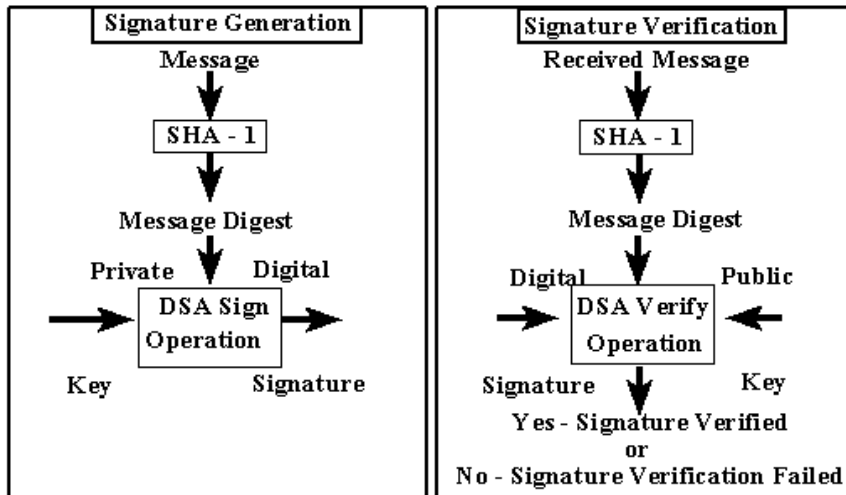


Fig. 1. The Digital Signature Standard

This leads to an important point, and one that's not obvious in figure 1—the association between a signatory and the public key. If the receiver of the message had obtained the public key via a separate e-mail, there might be no assurance that the public key was indeed that of the assumed signatory. An impostor could have generated his own key pair and sent the impostor public key in the e-mail, posing as the actual signatory. He could have then forged the message to his own benefit, used his private key to sign it, and sent message and key to the receiver who would verify the signature. Therefore, without a certified binding between a public key and its owner, the digital signature is ineffective. In practice, including a mutually trusted party solves this problem; the trusted party certifies the association of a signatory and public key and then distributes the certified public key as necessary.

In other words, a message is not authenticated until both the digital signature *and* the public key owner are verified. Of course, if an imposter discovers the *private* key without detection, then the system can be breached.

KODAK Picture Authentication Module Overview

The KODAK Picture Authentication Module architecture has three major components: authentication firmware on the camera, authentication software on a personal computer, and a simple public key management system to enable a trusted third party or third party proxy. In addition, a software API (application programming interface) is provided for integrating the digital signature verification and key management functions into a custom application.

Camera Firmware

In the KODAK Picture Authentication Module, a unique key pair must first be stored on each camera. The key pair is then used indefinitely; it can remain for the life of the camera or be replaced periodically as an additional security measure. A unique feature of the Picture Authentication Module is that the camera generates a unique key pair itself, without the need for another device. This increases security by confining the private key to the camera from the start. In contrast, if the private key was generated on another device and transferred to the camera, it might be stolen by intercepting the communication between devices.

Generating a non-deterministic, random key seed from a picture taken expressly for this purpose further increases security. The seed picture is taken during camera initialization and is destroyed immediately after the key seed is generated. Once the key pair is generated, the private key is hidden in the firmware, and the camera and authentication feature are ready for use.

Signing Pictures

The authentication firmware component signs the images. Referring to figure 2, Each image is JPEG-compressed by the standard camera firmware. Then the entropy-coded JPEG image data and the metadata are concatenated. The Secure Hash Algorithm is run on the concatenated data, generating the image digest. The digest and camera private key are input to the DSS Digital Signature Algorithm, which produces the digital signature for the image. The image file is written to the camera memory card, the digital signature and the camera's public key are written to the image file's authentication tag, and the process is complete. This process is repeated for every image captured, and cannot be disabled. Depending on the image size, the entire signing process adds only around 3 - 8 seconds to the standard image processing time for each image.

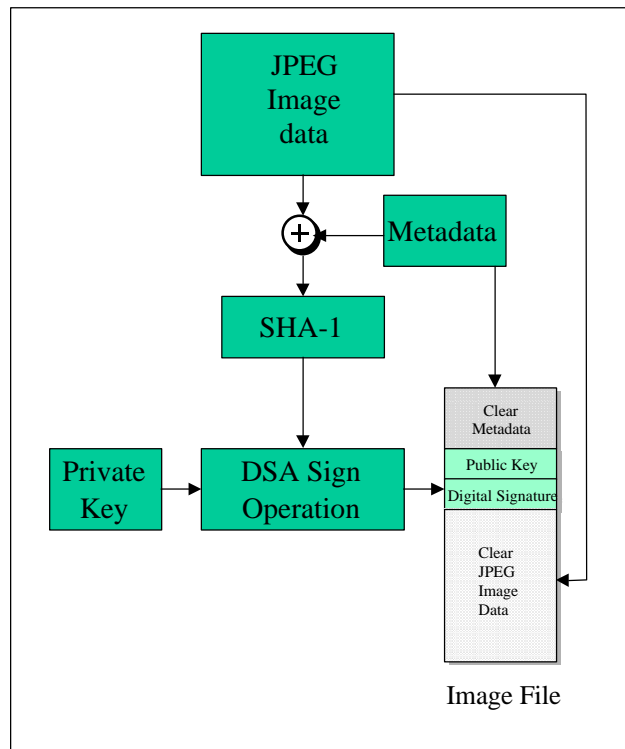


Fig. 2. Digital signature generation in camera

Image Signature Verification

The KODAK Picture Authentication Software was written to perform the image digital signature verification. As illustrated in figure 3 (Application Window), the Picture Authentication Software first allows an image or group of images to be selected and loaded from a file browser. Once loaded, image thumbnails are displayed in the main window. You can then verify the digital signatures from one or all of the images by selecting the images and clicking *Authenticate Pictures*.

Referring to figure 4, the software reads the digital signature and camera public key from the authentication tag, concatenates the image data and metadata (excluding the authentication tag), and inputs the concatenated data to the Secure Hash Algorithm. The new message digest, the

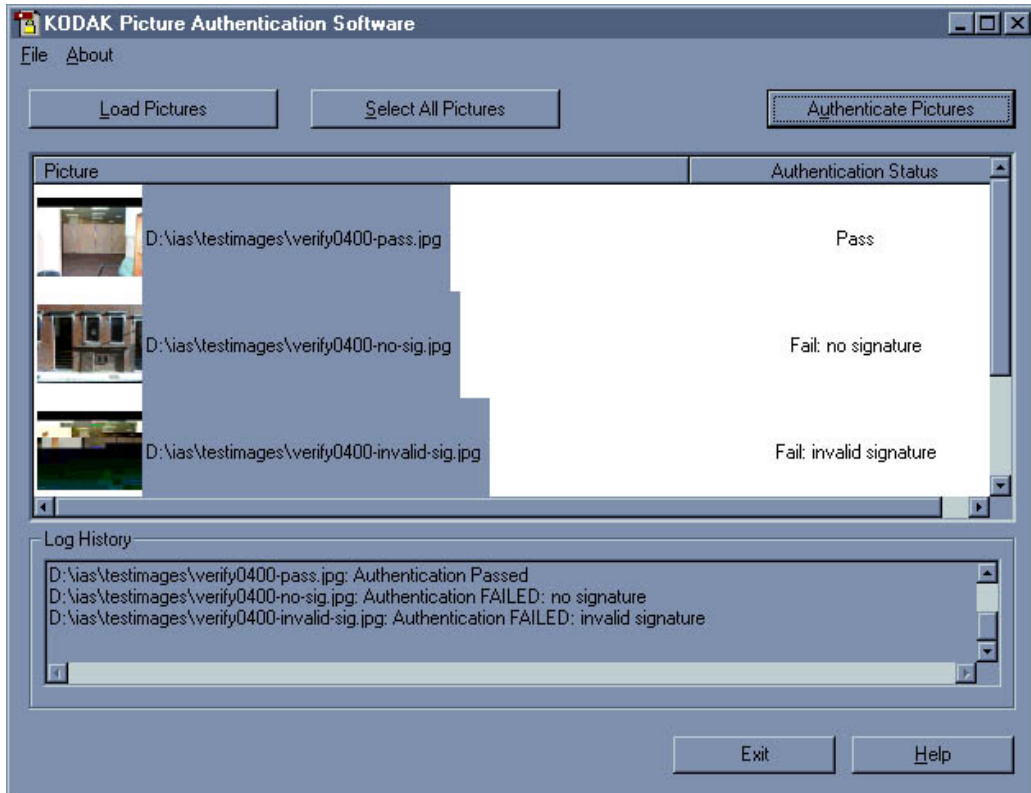


Fig. 3. KODAK Picture Authentication Software main window

camera public key, and the digital signature are input to the Digital Signature Verification Algorithm. The results of the algorithm are posted next to each image. If no authentication data can be found in the image file, the user is so notified.

Key Management

Verifying the image's digital signature is a first line of defense, especially for detecting image tampering by an unsophisticated attacker. However, digital signature verification alone does not constitute image authentication, since the image could have been intercepted, altered, and re-signed by a sophisticated impostor. Therefore, it is essential to know the possessor(s) of the private/public key pair used to sign the image and verify the signature, or at least to verify that the public key is legitimate and from a legitimate camera.

As described earlier, a trusted third party is usually involved in a public key infrastructure in order to "bind" the identity of legitimate users to their keys and distribute the keys as required. When someone obtains a public key from a trusted third party, they have a degree of confidence that the key actually belongs to the particular entity (person, camera), depending on the degree of certification and the policies and procedures of the third party. The KODAK Picture Authentication Module has a provision for such a third party, or at least a proxy thereof, by allowing one to record and save public keys from each camera.

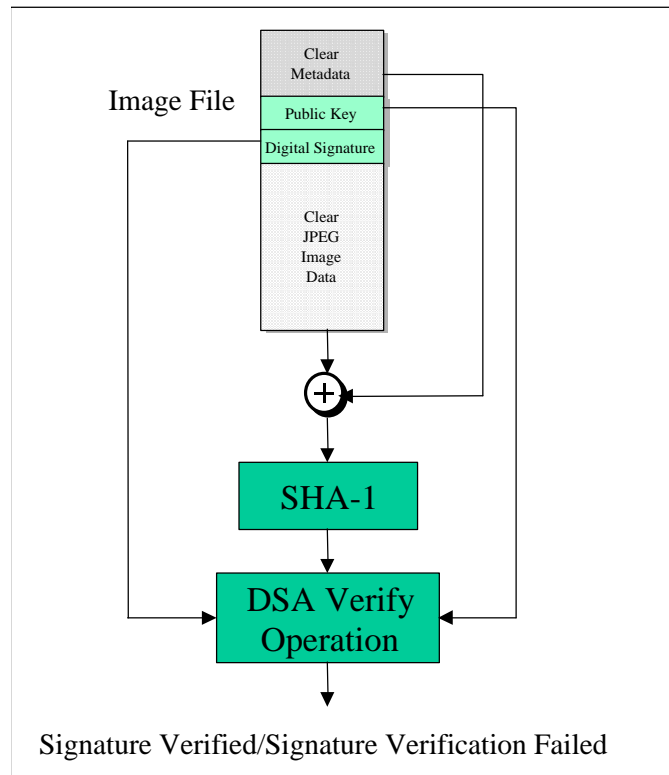


Fig. 4. Digital signature verification on host

Key recording is done as follows: after the camera generates the public and private key pair, an initial picture is taken and transferred to a computer on which the Picture Authentication Software is installed. The subject of the image is not important, since the file is simply a vehicle for the camera public key. The Picture Authentication Software allows you to extract the public key and some other pertinent data from the image file to a special key file. Key files from all of the customer's cameras can then be gathered and managed in any way the customer wants, from storage in a vault to management by a full-fledged certification authority. Because this process is vulnerable to attack, for instance an impostor inserting his or her public key as a legitimate one, it is imperative that the customer extracts and collects the keys in a secure location and then brings them to a secure destination.

Once the public keys are stored securely, the database can be consulted at any time to validate legitimate keys. How this is done in practice will depend on the customer, but the following scenario is illustrative.

Customer Use Scenario

The following use scenario illustrates how the KODAK Picture Authentication Module might be used in a larger organization.

The customer procures cameras and they are initialized at a dedicated, secure workstation by a trustworthy technician. After camera initialization, the technician takes the first picture for key extraction. The resulting image file, which is the usual JPEG file, is transported to the computer. From the Picture Authentication Software window, the technician loads and selects that first

image and extracts its public key. The extracted key data is recorded in the key file in the designated directory. As the customer receives and initializes camera shipments in this manner, the camera key files are collected by the technician and transferred to the secure key database.

Meanwhile, a third party software developer has written software that uses the digital signature verification and key extraction API's and has written automated image authentication software which integrates into the customer's secure image database infrastructure.

As users at remote sites upload images to the image database, the authentication software on a secure server analyzes each image. First, the public key is extracted from the incoming image file and then compared with the existing key database. If a match is found, the public key is considered legitimate; it belongs to one of the registered cameras in the field. The digital signature verification is then run to determine if the image was altered since it was signed at the time of capture. If the signature is verified, then the image is considered authentic. If the key does not match the ones in the database or if the digital signature does not verify, a message is generated and the case handled appropriately.

The image authentication system also allows authentication testing to be performed manually. This might be done routinely by a smaller organization or occasionally by a lawyer or administrator in larger organization if another party happens to challenge the authenticity of a particular image.

In the case of manual authentication, it is assumed that all aspects of transporting images and other data are done so securely, even if they are manual operations. For instance, the image is transported under appropriate custody to a standalone personal computer, for example, where a pristine signature verification application resides. An authorized copy of the key database is also brought to the computer under careful custody.

An authorized individual launches the application, locates the image, and runs the verification test. The application verifies the image's digital signature using the public key from the image file.

Next, the user needs to know if the public key on the image file is a legitimate one. The user selects the key extraction function and extracts the public key to a new key file on the computer. The user then opens the file with a text editor and observes the public key value. Using a standard program to search files for text strings, the user searches the key database for the key value extracted from the file. A match is made. The user has now determined the key is legitimate and the signature is verified. The user is now confident that the image is indeed authentic.

Security Considerations

While the KODAK Picture Authentication Module provides the components needed for reliable picture authentication, the customer must deploy the system with the proper security measures. While detailed security instructions are beyond the scope of this document, it is assumed that the technical details disclosed provide sufficient information for qualified persons or groups to architect, deploy, and administer their picture authentication system properly.

Camera Custodianship

However unlikely, a camera is somewhat vulnerable to attack. An unsupervised camera could be reverse-engineered, revealing its private key. Possession of a legitimate private key by an impostor can allow images to be forged without detection. Therefore, proper custody of each camera is absolutely critical throughout its service life, beginning immediately upon receipt of the camera unit(s)

Initialization

As described earlier, the unique private/public key pair is generated on the camera at initialization time. Trusted individuals should perform this initialization in a secure and properly supervised location.

Key Extraction

Once each camera is initialized and the public key is extracted, the public key should be transferred immediately from the camera to a secure file system* according to the product instructions. This is a vulnerable step, since this is where an impostor could enter an illegitimate public key to the collection of legitimate ones. Thus, key extraction must take place in a secure location on secure computing equipment. It is extremely important that the public keys be in secure custody from the time of extraction to their final destination.

*In this step, the file system must not be accessible by unauthorized parties. It could be behind a secure firewall, or it could be simply on a properly initialized and supervised standalone (non-networked) personal computer.

Camera Use

Once the public key is extracted from a camera and stored properly, the camera can be put into service. Proper custodianship must continue as the camera is transferred to users. For example the camera must be used only by authorized individuals and either properly supervised or stored between uses.

Authentication

Picture authentication must also be performed in a secure location, since it is possible that an impostor could modify the authentication software to give incorrect results. The Customer Use Scenario described earlier covers two authentication procedures. One is done routinely as part of an automated process and the other is done on a standalone computer. In both cases, the following two important security issues must be understood.

Recall that an image can be considered authentic only if both the digital signature and the public key are verified. Referring back to figure 3, note that the digital signature verification is performed using information carried in the image file. To alter the results of the digital signature verification, an attacker could simply alter the signature verification software on a computer. Thus this process must occur on a secure computer. The public key verification can only occur by comparing the public key in the image file with the public key in the key database. Not only must the key verification software be protected, the public key information must be retrieved from the database in a secure fashion, regardless of whether the key information is transmitted across a network, or by hand on a diskette.

Decommissioning Keys

It may be prudent to periodically decommission key pairs. Decommissioned public keys would be transferred to a separate, secure database, with a record of the decommission date for each key. Reinstalling the firmware and reinitializing each camera would generate new key pairs (the decommissioned private key is destroyed in the process). If routine decommissioning were practiced, each key pair would be valid only during the life of the key pair on the camera. In some cases, periodic decommissioning and regeneration of new keys could thwart reverse engineering of cameras to obtain private keys. Because keys are short-lived and valid only for the life of the key pair on the camera, reverse engineering a camera to discover its private key is less advantageous. Illegitimate use of a stolen private key could be detectable once the key pair had been decommissioned.

Summary

Through proper use of authentication-enabled digital cameras and key management techniques, businesses can greatly improve the security of images that move through their workflows. The DSS algorithm provides a trusted, secure method for enabling this process.

References

1. U.S. Department of Commerce, National Institute of Standards and Technology FIPS PUB 186-1, <http://www.itl.nist.gov/fipspubs/by-num.htm>