Technical Disclosure Commons

Defensive Publications Series

January 02, 2019

Verifying authenticity of digital images using digital signatures

Neil Dhillon

Tanmay Wadhwa

Verifying authenticity of digital images using digital signatures

ABSTRACT

This disclosure describes techniques to verify the authenticity of digital images using digital signatures. A secure element on a client device generates a public-private key pair and the public key is uploaded to a server. When a new image is captured by the client device, the device hashes the image and signs the hashed result using the private key to form a digital signature. Both the original image and digital signature are uploaded to the server. The server hashes the original image and compares it to the digital signature that has been decrypted using the appropriate stored public key. A match indicates a verified image and the server provides an indication of the verified status of the image to a requesting device along with the image. The described techniques thus enable images to be authenticated as to source and content.

KEYWORDS

- digital signature
- image authentication
- image source
- source verification
- image forgery
- cryptographic processor

BACKGROUND

Machine learning techniques (e.g., deep learning) and neural networks have enabled more advanced recognition of image content as well as generation of images by machines. Using techniques such as Generative Adversarial Networks (GANs), photorealistic images and videos can now be generated and presented convincingly as captured images and videos, thus presenting new ethical and social challenges. One example is “deep fakes,” which are images or videos synthetically generated. An example technique to generate deepfakes is to superimpose existing images over other images and videos.

Technical breakthroughs in machine-learning enable malicious actors to cause harm using deepfakes. For example, such techniques have been used to generate fake videos with celebrities, “fake news” videos, malicious hoaxes targeting political figures, etc. Furthermore, an increasing number of users gather news and other information from social networks, and such information is often in the form of images that have been shared on these platforms. The need to defend against misinformation provided in modified images and videos has increased the need for automated techniques to verify the source and integrity of images and videos, e.g., to obtain confirmation that the content of the images or videos has not been modified after being published by its source.

DESCRIPTION

This disclosure describes techniques that verify the authenticity of digital images, including photos and video, using digital signatures. Described techniques provide authenticity guarantees to images captured by various user devices and shared on server platforms, verifying the integrity of the images. Additionally, the source of the images can be identified and authenticated.

Generating cryptographic keys for client devices

Described techniques include the use of a client device and a server. The client device includes a secure element, which is a tamper-resistant hardware platform, e.g., that includes a secure cryptographic processor. The client device can be, for example, a smartphone, camera, or any device that can capture images (e.g., photos or videos).

Public-key cryptography is used to digitally sign and authenticate images captured by the client device. The secure element on the client device generates cryptographic keys including a public and private key pair using a cryptographic algorithm such as RSA. The private key never leaves the secure element on the client device. The public key is uploaded to a server used in the described techniques as detailed below. For example, the server can belong to a photo sharing service, social networking service, messaging service, or other service that allows access by devices to images uploaded to the server.

For example, generating and storing the cryptographic keys can be performed at a factory or other place where the client device is manufactured or assembled. For example, the manufacturer of a device can upload the generated public key to the server. In other examples, the cryptographic keys can be generated and stored when a user first signs into (e.g., creates an account with) an image/video management application on the client device, or opens an account with a photo sharing service or other service that can host images. The generated keys are tied solely to the particular client device, or the generated keys are tied to both the client device and to the user's account with the service.

The server receives the public key generated by the client device. In some examples, the server also receives metadata that is associated with the public key, e.g., metadata descriptive of the client device (e.g., the model or type of client device), descriptive of the public key (e.g., the date of public key generation), and/or descriptive of the user account (e.g., username or other information, if permitted by the user). The server stores the public key is association with the metadata.

Authenticating images captured by the client device

When the client device (with a cryptographic key stored onboard) captures images (e.g., photos and/or videos), such captured photos/videos are digitally signed for authentication, prior to upload to a server or to other devices using techniques described herein. Fig. 1 illustrates an example technique for this process.

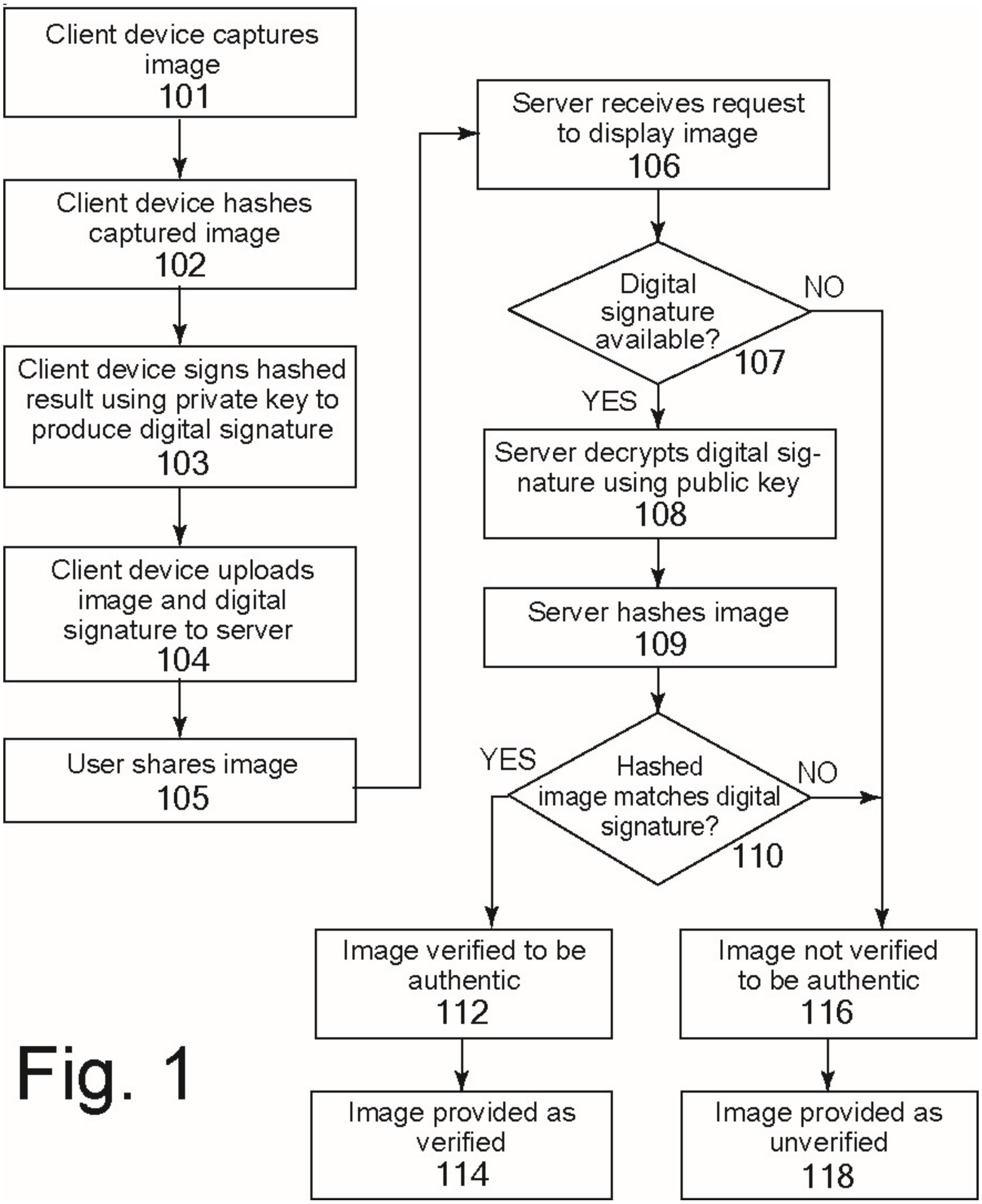
Client device captures image
101
Client device hashes captured image
102
Client device signs hashed result using private key to produce digital signature
103
Client device uploads image and digital signature to server
104
User shares image
105
Server receives request to display image
106
Digital signature available?
107
NO
YES
Server decrypts digital signature using public key
108
Server hashes image
109
Hashed image matches digital signature?
110
YES
NO
Image verified to be authentic
112
Image not verified to be authentic
116
Image provided as verified
114
Image provided as unverified
118
Fig. 1

Fig. 1: Example method for image verification

The client device captures (101) an image (e.g., a photo or a video) that is to be attested for authenticity. For example, the user of the client device provides a command (e.g., a shutter click) to the client device to capture the image. The client device hashes (102) the captured image using a hash function, e.g., SHA256 or other function. The client device signs (103) the hashed result using the private key stored on the secure element of the client device to produce a signed hashed image, which is the digital signature.

With user permission, the client device then uploads (104) the (original) image and the digital signature to a server that hosts the image for network access by other devices (or sends directly to other devices). Other information, e.g., image metadata, is also uploaded, if permitted by the user. In some examples, the server can be part of a photo sharing service or other service. The server stores the image in association with the digital signature. In addition, the server associates the image with the public key associated with the client device (and/or user account) that uploaded the image. The public key for the client device is available to the server, e.g., as provided by the device manufacturer, as described above. Alternatively, the public key, which is locally available on the user device, can be uploaded with the image.

The uploading user then provides a command to share (105) the image, e.g., such that a link to the image is shared with one or more other users and/or user devices, or provides the link to a website, social network, or other accessible service or device. In some examples, the image can be shared with all or a subset of users of the photo sharing service, can be available for public viewing by any device that can access the server, etc.

At a later time, the server receives a request (106) from a device (e.g., a requesting device) of a user that has been allowed access to the image. The request is to retrieve the image, e.g., for display on the requesting device. For example, the shared link to the image is displayed in an application of the requesting device, and a user of the requesting device selects the link to request the photo sharing service to send image data to the requesting device for displaying the image on the requesting device.

Upon receiving the request for the image, the server checks (107) if a digital signature is available for the requested image. For an image captured and associated with a digital signature as described above, the digital signature is stored on the server in association with the image. If a digital signature is not available, the server indicates that it is unable to verify the authenticity via described techniques (116), and the image is tagged as unverified and is provided as unverified (118) to the requesting device for display.

If a digital signature is available, the server decrypts (108) the digital signature using the stored public key associated with the client device (and/or user account) that uploaded the image. The server hashes (109) the (original) image using the same hash algorithm as the secure element used to create the digital signature. The server compares (110) the hashed image to the decrypted digital signature.

If there is a match, the original image is verified as authentic (112), e.g., the source and integrity of the image are determined to be verified such that the image is the same as the original image that was uploaded from the client device. The image is tagged as verified, and is provided (114) as a verified image to the requesting device for display. Additional information related to the image can also be provided by the server to the requesting device (if permitted by the user that captured the image), examples of which are described below.

If the server-computed signature doesn’t match the stored signature, the original image is tagged (116) as unverified. In response to the request, the image is provided (118) as an unverified image to the requesting device for display.

In various implementations, verification of the image using the digital signature and the public key (e.g. 107-112 and 116) can be performed at different times or multiple times, e.g., upon receiving the uploaded image at the server, periodically by the server, etc. Furthermore, other devices (e.g., servers that provide other services, client devices of requesting users, etc.) can perform the verification if the devices have received the digital signature for the image and the public key for the user device.

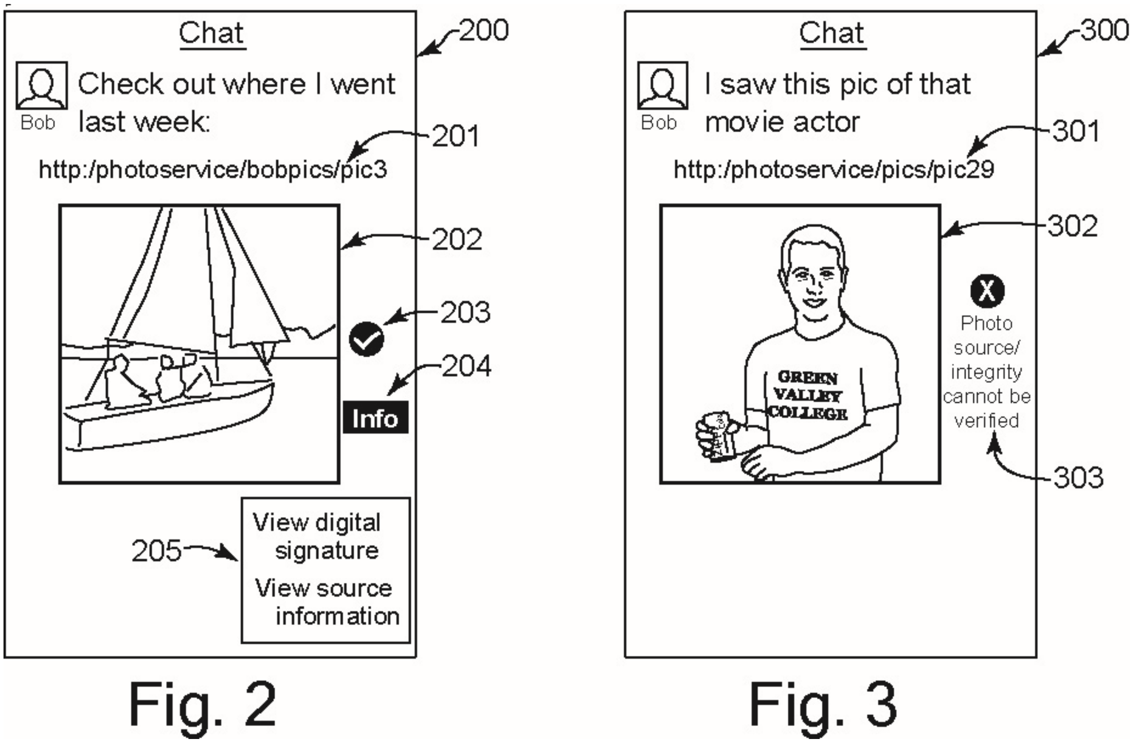
Chat
Bob
Check out where I went last week:
http:/photoservice/bobpics/pic3
200
201
202
203
204
Info
205
View digital signature
View source information
Fig. 2
Chat
Bob
I saw this pic of that movie actor
http:/photoservice/pics/pic29
GREEN VALLEY COLLEGE
X
Photo source/ integrity cannot be verified
300
301
302
303
Fig. 3

Fig. 2 and 3: Example user interface showing verified/unverified images

Figures 2 and 3 illustrate examples of images displayed by a requesting device, where the images have been verified and unverified as authentic using described techniques.

In Fig. 2, a user interface (200) displays a chat or messaging conversation, in which a user has provided a link (201) to access a first image. The user of the user interface has selected the link, which causes a first image (202) to be downloaded from a server of a photo sharing service and displayed in the user interface. The first image was uploaded by a different client device using the techniques described above, e.g., with a digital signature and having an associated public key for the client device accessible to the server.

The user interface also displays additional information related to the image. In this example, the image has been verified by the server to be authentic, so the additional information includes a user interface element displayed on or adjacent to the image to indicate that it has been authenticated. In this example, a checkmark (203) is displayed adjacent to the image. Other such elements that can be displayed include a watermark on the image or other marker that indicates that the image has been authenticated.

An information button (204) (or other user interface control) is also displayed adjacent to the authenticated image. If this button is selected by the user (or other appropriate control selected), a menu (205) of items of additional information is displayed as shown. An item can be selected by the user to cause the associated information to be displayed in the user interface. In this example, the available menu items include an item to cause the digital signature of the image to be displayed, and an item to cause source information related to the image to be displayed. The source information can include the metadata information related to the user device or user account used to upload the image, such as the model of user device, date of upload of the image to the server, etc. The source information is displayed with permission of the user that uploaded the image.

In Fig. 3, a user interface (300) displays a similar chat or messaging conversation to Fig. 2, in which a user has provided a link (301) to retrieve a second image. The user of the user interface has selected the link, which causes a second image (302) to be downloaded from the server of a photo sharing service and displayed in the user interface.

In this example, the second image has not been verified to be authentic by the server hosting the image. For example, the second image may not be associated with a digital signature and/or a public key accessible to the server, or the server's hash of the image may not have matched an associated digital signature. To indicate the lack of verification, an indicator (303) is displayed adjacent to the image, which in this example includes an icon and text indicating that the authenticity of the source and the integrity of the image cannot be verified. In some implementations, users of requesting devices can select preferences for the device to not display unverified images.

Described techniques provide multiple benefits. One benefit is that authenticity and integrity assurances can be provided for image content that is captured by user devices and uploaded to photo sharing services and other services. Described techniques can also be beneficial for image search features. For example, users may use an image search function to check if a particular image has been posted at sites or services without being modified, or to find sources of the image. In addition, metadata for images used by the described techniques can provide additional criteria by which to search for images.

The techniques described herein are implemented upon specific user permission to access and process a user's images. Only images, videos, and image/video metadata for which the user grants permission are processed with the techniques described herein.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user's activities, social network, or social actions, profession, a user's preferences, a user's devices, or a user's current location), and if a user device is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

This disclosure describes techniques for detection of the authenticity of an image to be downloaded, i.e., verifying that the image has the same image content that originated from the source without modification. A secure element on a client device creates a digital signature for an image using a secure private key. The digital signature is used to verify that an image on a server has originated from that client device and the integrity of the image has been maintained. Images that have been so verified are indicated to the downloading user, thus providing assurance to users that a displayed image is authentic and helping reduce the spread of misinformation enabled by image forgery and the creation of fake images.