System Design INL1 Written

Casper Kristiansson

March 2, 2024

1 Introduction

The goal of this assignment is to propose a way of developing a new system that can distinguish AI-generated and camera-recorded photographic images. In the assignment, it will be discussed everything from assumptions taken into account to the architecture of it.

2 System Assumptions - *Components and their resources*

To design such a system we need to make a few assumptions for both cryptographic algorithms, software, and hardware of the devices involved. First off we assume that the cryptographic algorithms that are used to sign the images have no current way (easy way) of being attacked. We also assume that each camera hardware can generate a key pair securely. The private key should securely be stored within the camera hardware and cannot be extracted or duplicated by any external users. We must also assume that the image sensor can sign the images taken. The sensor should be able to sign them directly using any processing of the image happening before it. Lastly, we need to assume that any image/video player should be able to validate signed images.

3 User Assumptions - *Their goals and capabilities*

For this system, we also need to make a few assumptions for the users of this system. First off the users should have general knowledge of cryptographic signatures and how they can be used to verify the authenticity of images. This is especially needed for the user to understand the warning an image viewer/video player will show if a signature is missing or invalid. The user also needs to be able to trust the system and that the system processes are secure and can distinguish between AI-generated and camera-recorded photographic images.

4 Adversary Assumptions - *Their goals and capabilities*

The adversary of this system probably has the primary goal of creating and distributing fake images or videos. The fake material will be made so that they are indistinguishable from real ones. Furthermore, we can make assumptions about the adversary such as the individual has a wide range of capabilities such as the resources of performing cryptographic attacks but are limited to today's technology. We also need to assume that the attack could be an individual who has insider knowledge and access to camera manufacturing companies.

5 Requirements - *e.g., Security properties, authentication, performance*

When designing this system we will need to make sure that it upholds a few requirements. For the security of the system, it first needs to have good integrity where it can make sure that images/videos cannot be altered without being detected. This is crucial if the system should be able to prevent adversaries from modifying the content. The system then needs to be able to authenticate and verify the images/video's origins such as the specific device model and manufacturer. For the authentication process, the system will use digital signatures to sign the camera's images and video using the sensor (processing unit with the sensor). This means that each camera sensor needs to be able to perform basic cryptographic operations and have a unique key pair assigned to it.

6 Architecture - *e.g., Server and smartcard or smartphone, what is computed where*

The architecture of this system will involve a lot of different devices. During the manufacturing part of the image-capturing devices, we need a trusted Certificate Authority (CA) [1] to issue a digital certificate for the device. The goal of this is to bind the camera with a certificate that contains a public key. Then we will have cameras act as a device where it has a sensors that can perform cryptographic operations and sign images. This means that the camera/sensor

will capture an image, sign it, and later store it on the device. The second part of the system architecture is the image/video viewing software which has the goal of being able to verify signatures using the certificate attached to the image.

7 Data at Rest - *How and where is it stored*

The data associated with this architecture is the signature generated for each video/image and the certificate for each device. First, the signature generated from the video/image will be either stored directly in the image/video file or the metadata of it. By doing this it would allow that if an image/video would be moved to another device the signature would still be intact. The certificate would be stored in the same way. Doing this would allow an image/video viewing application to directly access the signature and certificate to verify the image's integrity.

8 Data in Transit - *What is sent and how*

The data in transit for this system will mainly first consist of the image and its signed signature. Each image will also contain a digital certificate for the specific device. Both the signature and the digital certificate can either be embedded in the image or attached as metadata. The second part where data will be transmitted will be during the manufacturing phase of the device, where a digital certificate will be issued. This process involves creating a Certificate Signing Request (CSR) [2] that will contain the public key of the device, device information (serial number, manufacturing company), and a digital signature to verify that the public key is associated with the device (generated using the private key). The Certificate Signing Request will then be sent to a Certificate Authority which will issue a digital certificate for the device.

9 Cryptographic Algorithms and Protocols - *e.g., AES, RSA*

The algorithms and protocols that will be used in this system are divided into three different parts. First of we need a Digital Signature Algorithm which will be used for signing the images/videos generated. A good cryptographic protocol for that would be RSA which is a good algorithm for both creating and verifying digital signatures. Second of all, we need a way to hash the image data before signing it so that it can later be verified. For this, we can use SHA-256 which is a good secure hashing algorithm. Then for the digital certificate, we will be using a Public Key Infrastructure for managing the certificate. This means a wide range of protocols will be used to keep track of current certificate statuses etc.

10 Implementation - *e.g., libraries*

As for the actual implementation of this system, we first need the technology where each camera has a system that can perform cryptographic operations. This system will have to be of the type hardware security module (HSM) so that it can be temper-proof. Later the keys generated will be stored in the secured hardware, which will ensure that the private key cannot be extracted by an attacker. The certificate issued by a certificate authority will also be embedded in the camera module so that it can be attached to each image/video generated. As for the software required for the cryptographic algorithms and protocols the library OpenSSL will be able to cover the required functionalities because it supports the needed protocols (RSA, SHA-256) and can generate keys.

11 Parameters - *e.g., key length*

For the parameters of the architecture, it will mostly be linked to the actual key length for RSA. Because each image/video will have to go through the process of being hashed and signed we would want to find a balance between security and performance. When finding a good balance there are different things we need to consider. First off I believe the encryption phase needs to be fast but the decryption phase could take a bit longer of time. Therefore using a key size of either 2048 or 4096 would be a good balance between encryption and performance.

12 Ensuring Randomness

For this system, we would need randomness for both its security but also to avoid collision when signing images/videos. First of we would need randomness during the key generation phase. This is to ensure that the private key generated is unique and therefore secure. As for the signing phase of the images/videos, we would need to generate a random number to sign the file. The randomness will make sure that the signature won't involve collisions. Adding randomness to the signature would help with security by making it harder to figure out the private key.

13 Trade-offs and Risks - *e.g., privacy vs. utility, cost vs. security*

First off regarding the performance and security of the implementation, I mentioned in section 12 that there will have to be further research into finding the best performance and optimal security in both the protocols used and also in the parameters used (key length). During the manufacturing phase of the cameras, each device will be issued a digital certificate for the device. In section 8 it is mentioned that each certificate will contain information such as the camera type, serial number, and manufacturing company. While this information is not required it does expose some privacy information of the user. A second important part of privacy concerned with this system is that any person could tell based on both the signature and certificate of an image if another image has been taken with the same device. This means that users can be able to track multiple images back to the same device which may expose privacy information. Regarding the security of the device with the trade-off being the cost this is especially an important part. To implement this system each camera would need a new system that can perform cryptographic operations and has to be a hardware security module (HSM). This will be required to make sure that the device will be tamper-proof but it will add extra cost for manufacturing the device.

14 Legal Considerations

As for the legal considerations when implementing this system, the architecture for generating and storing information has to be able to comply with regulations such as GDPR. GDPR specifies in Europe a user has the right to access, and erasure, and that the data is restricted, etc. This means that the software and hardware will have to comply with these regulations. This is especially true in regards to the serial number attached to the certificate but also the signing processes. A user should be able to specify if they want to sign images or not on the specific device.

15 Usability Considerations

For the usability of the system, it will have to be simple for the user and it wouldn't require a lot more actions to be performed when for example taking images. Such options might be turning on/off the signing of images/videos captured from the device. Second of all the software used for verifying the image's integrity and if they are AI generated or not will have to be user-friendly. This means that a non-technical person understands what the verification process means and why it can be trusted.

16 Non-Cryptographic Threats and Countermeasures

The biggest concern with this system is the noncryptographic threats to this system. In this example, we will see the attacker as someone who would want to be able to label AI-generated images as images taken by a camera. For this, we first have physical tampering of the device. Because each device will have its private key stored on it an attacker might want to try and extract the key to then use that key to sign other images. For this reason, I decided that each camera should have a hardware security module that should be tampered-proof (section 10). This means that if an attacker tries to extract the key on the device the system will handle and perform countermeasures such as deleting the key stored on it. The biggest flaw of this system is concerning the manufacturing part of the cameras. The cameras manufactured by the manufacturing company will get a digital certificate. This means that if an attacker works for one of these manufacturing companies they might have the knowledge and access to be able to generate certificates for a nonexistent camera. This means that if an attacker has this power the entire system will become useless. A countermeasure for this would be to only allow the biggest camera manufacturing companies to generate top-graded certificates where they can prove that the process of manufacturing the cameras is safe and secure. Manufacturers that are not able to prove this could get a lower-graded certificate (when viewing an image they get a warning of being potentially AI-generated).

17 Cryptographic Threats and Countermeasures

For the cryptographic threats, we first have the keys generated for each device. If the keys were exposed it would defeat the purpose of the system. The keys could be insecure if the device is not using a hardware security module for managing the keys. Second of all if the keys or signing the images could be attacked (signing images that are not yours). If these algorithms and protocols could be attacked it would render the system unable. Also, an important part of this would be that if the system is not implemented correctly it could introduce vulnerabilities. This for example could be implementing cryptographic algorithms wrong or implementing a randomness algorithm wrong/using a library wrongly.

References

[1] *What is a certificate authority (ca)? - ssl.com*, https://www.ssl.com/article/what-is-a-certificate-authority-ca/, (Accessed on 02/29/2024).

[2] *What is a certificate signing request (csr)? do i need one?* https://www.globalsign.com/en/blog/what-is-a-certificate-signing-request-csr, (Accessed on 02/29/2024).