Photo Encryption in Medicine

Georg Bernold, Gabor Szivos

Index Terms—Photo Encryption, Image, Encryption, Medicine

1 Problem Description

Privacy is becoming more important day by day. Numerous regulations and laws require it across all kinds of use cases and industries. A well known way to achieve privacy is through encryption, bla bla bla define encryption maybe insert source?. As encryption requires computational resources, its performance is dependent on the platform it is used on. To fullfil the requirements of privacy in photography, it might be necessary to encrypt photographs right after they are taken.

Especially in the context of medicine this is very important as health data is considered the most private data fill reference here. Medical photographs show a patients body, they are used to document visual properties of the skin, the state of a wound or the progress of a plastic surgery. All of the above mentioned photographs document very private information about an individual, which is only to be seen by authorized persons.

Another important application for encryption of photographs is journalism. Journalists may take pictures in

This demo file [?] is intended [?] to serve [?] as a "starter file" for IEEE Computer Society journal papers produced under LaTeX using IEEEtran.cls version 1.8 and later.

2 Expected Results

Our goal is to evaluate different possibilities, to make on-the fly encryption for Digital single-lens reflex (DSLR) cameras possible. Our final goal is to create a method which can evolve to a best practice, which gives the medical and journalist community the possiblity to protect their data with the help of encryption.

To make this happen we found four different solutions which could be used to make DSLR camera encryption possible. For Canon cameras there is an open source project called Magic Lantern, which enables the extension of the stock firmware with new functionalities, like encryption or better autofocus. This project already has an encryption module, however this module uses relativle weak algorithms. It also makes possible the use of stronger algorithms (RSA), however these are very slow. Our plan is to create a new encryption module with better and faster encryption algorithm (like Chacha20 with Poly1305). During this task we are going to learn about how the encryption in Magic Lantern works, and how different kind of algorithms perform. Additionally we also going to test, if there is a possibility to implement this feature to enable the plausible deniability. Plausible deniability, according to the Oxford dictionary, ist "the possiblity of denying a fact, especially a discreditable action without arousing suspicion".[1] It worth mentioning, that the plausible deniability, is not really important in the medical context, however it might help for the journalist community.

As a second method we found, that there is way to implement a special SD-Controller, which is connected to a field programmable gateway array (FPGA), which is again connected to a mass storage. In this setup the FPGA is responsible for the encryption process. Our goal is to find out how such a special SD-FPGA interface performs in a DSLR camera. And how the encryption algorithm performs on an FPGA. And as in the previous setup we are going to try to evaluate if there is a possibility to implement such a system in a way, where the user has an option for plausible deniability.

The third option would be to use a special WiFi capable SD-Card which in turn would send the photos directly to the users smartphone (or other smart device), which would take care about the encryption. The questions we ask regarding this method, and try to answer is how to transform images from the DSLR camera to the smart device and how to delete securly the data after the transmission was done. We are also going to evaluate the preformance and the possibility to implement the option for plausible deniability.

The last method we will consider is to place a whole PC between the mass storage and the camera in the form of an Intel Edison, which a fully functional PC in the size of an SD-Card. As in the previous setups we will evaluate the performance and the option for plausible deniability.

2.1 Subsection Heading Here

Subsection text here.

2.1.1 Subsubsection Heading Here

Subsubsection text here.

3 Method

To be able to evaluate and compare the results, how the previously mentioned methods work, we are going to create for each of them a prototype. This means we are going to write a module for the Magic Lantern project, which will replace the current implementation with a stronger encryption algorithm. Based on the work of Davidsson et. al. [2] we are going to create a modified SD-Card which includes an FPGA modules between the SD-Card controller and the mass storage. As what the smartphone setup concerns, we are going to write an android app, for handling the datatransfer between the DSLR and the smartphone, with the help of an SD-Card with on-board WiFi capabilities. For the Intel Edison solution, we plan to use the General Purpose Input Output (GPIO) ports to fake an SD-Card for the DSLR-Camera. And to implement the encryption feature we are going to write a hook for the GPIO ports, meaning if there is a request from the camera to save an image on the mass storage the interconnected Intel Edison will the image and encrypt it, and on the other side it will send back a response for the camera, that the image has been successfully saved.

The decryption of the files, apart from the smartphone solution, where the smartphone takes care of the crypto, will be realized as an external program which is independently runs on a desktop PC, or any similar architecture. This means there will be no on board solution for the decryption, because this would jeopardize the goal of the encryption.

The planned steps for realizing this project would be the following. First we are going to perform a state of the art analysis to get a better view about the current situation regarding the hardware and software solutions of encryption in the field of embedded systems. As the software solutions seem to be easier to accomplish, firstly we are going to implement the Magic Lantern modul. Following that we will implement the smartphone solution with the WiFi-capable SD-Card. Next we will take care of the implementation of the Intel Edison solution. And last but not least we are going to implement the FPGA tweaked SD-Card.

4 References

References

[1] plausible deniability - oxford dictionaries. Oxford dictionaries - plausible deniability Online; Accessed: 30.03.2017.

[2] Alexander Davidsson and Torbjrn Rasmusson. A vhdl architecture for auto encrypting sd cards. Master's thesis, 2016. 36.