Glitter PUF: A Passive Physical Anti-Tamper PUF Based On Images Of Glitter Reflections

Noeël Moeskops

Delft University of Technology

TU Delft

Glitter PUF: A Passive Physical Anti-Tamper PUF Based On Images Of Glitter Reflections

by

Noeël Moeskops

Supervisor:	Dr. ir. Mottaqiallah Taouil
Committee:	Dr. ir. Sten Vollebregt
Project Duration:	November, 2023 - October, 2024
Faculty:	Electrical Engineering, Mathematics and Computer Science
Department:	Quantum & Computer Engineering

TU Delft

Summary

The importance of cybersecurity is ever more increasing. And with that, the importance of anti-tamper technologies. Physical anti-tamper technologies have existed for a while, but passive (battery-less) approaches are lacking. Some solutions exists ([1], [2] and [3]). But are sparse, complex, expensive and only cover a small portion of the device. The subject of this research is that of Glitter Physical Unclonable Function (PUF). A passive physical anti-tamper technology that protects the whole device, and is technology independent. It works as follows:

The secure component (a Printable Circuit Board (PCB)) is placed in an empty box. A camera and Light Emitting Diode (LED)s are added to the center of the PCB. Then a mixture of glitter and transparent resin is poured into the device (including the camera and lights). The resin hardens and the box is closed (being light tight). Enrolment: A picture is taken with the camera inside the box with the LEDs turned on. The light bounces throughout all the glitters. This picture is used to encrypt the critical data on the PCB. When a tamper has occurred (by pruning the enclosure) the glitters are destroyed or irreversibly moved. Thus the picture taken afterwards is different. Authentication: Another picture is made. If the picture is within a certain threshold (set in enrolment), being the maximum allowed colour distance for each pixel to the reference picture. Then the picture can be used to decrypt the data. Otherwise, access to the data is lost.

For encryption and error correction several algorithms have been researched and tested. Especially the Error Correcting Code (ECC) used in biometrics are interesting, since they have the same criteria as glitter PUF (using real word data as a key); input data not always being the same. For the protoype Fuzzy Commitment is implemented, being the baseline for biometric encryption. A prototype of the glitter PUF has been made using a Raspberry Pi (RPi) camera and a 3D printed case. This prototype is used to evaluate certain characteristics such as how it behaves in time, temperature difference, ageing and what picture settings are most optimal. These characterizations are important to know how the device behaves when in operation. As they must deviate from the results gathered when an actual tamper has been taken place.

The ageing and temperature tests show that the sytem is susceptible to temperature cycles, but when the same temperature as the enrolment is reached the difference is lessened. Furthermore, the system seems to stabilize and not increase in error after about 23 cycles. After doing so many cycles it can be calculated which (sub)pixels are the most stable. Choosing only the most stable (sub)pixels for enrolment/authentication will reduce the error introduced by temperature and ageing. Drilling shows a linear increase in (sub)pixel distance from the reference. The drill displays hotspots, whereas the temperature tests (mainly) show noise in the red channel. So an algorithm that considers these differences must be implemented. An implementation using binning for error correction, fuzzy commitment, segmenting the picture in small blocks, using Adaptive Equalization; grey-scaling, and taking only 20% of the most stable (sub)pixels, results in 5% of the original picture data while still containing enough information to reliably detect drilling's.

Contents

1 Introduction

This section introduces the topic of the thesis. Its motivation, why a new passive anti-tamper method is nessesary; the current state-of-the-art (Section 1.2) anti-tamper methodes (both passive and active), it explains PUF and the contributions (Section 1.5) that this thesis contributes (Section 1.4). Furthermore, the document structure is described to allow for easy reading.

1.1. Motivation

Cybersecurity is of ever-increasing importance in the modern digital world. Many systems the world relies on such as cloud computing, bank transactions and information distribution are fully automated servers. Even more so, vital infrastructure such as public transport, power network and air-traffic etc. rely heavily on computer systems. These systems are the subjects of ever more increasing attacks by Advanced Percistance Threat (APT)s ([4], [5], [6]). Proper security is therefore in order. However, the main focus in the security industry is focused on the digital domain. On many systems once physical access has been reached the device is at the mercy of the actor. Physical attacks account for more than $1 trillion loss in revenue in 2022 [7]. Several ways to mitigate physical security exist ([8], [9], [10]), but are mostly specific on a certain device, situation or a threat local to the technology. An uniform system which is able to detect physical attacks regardless of the underlying technology is more beneficial. As the underlying technology is then able to freely change, and innovate without compromising performance for security. Because its more flexible, scaleable and generic solution that doesn't require the system underneath to be altered significantly.
Physical security can come in many forms (as discussed in the next section). One aspect that is greatly lagging is that of passive (i.e. battery-less) systems. This kind of system is especially useful when the device needs to be secure when in transit or stored for a long duration. Especially for vital infrastructure industries (like power, public transport or air-traffic) this is of importance. Since actors with bad intent may wish to tamper with vital equipment and cause denial of service or extract sensitive information. The lifetime of this equipment may be long, and their location ever-changing. Especially in transport it's difficult to ensure that a system has not been tampered with. An example of these systems could be a handheld device (phone, tablet, storage devices etc.), critical infrastructure (computer servers, security gates) or to protect proprietary firmware components in vehicles.
In-order to mitigate the issues stated above. A form of anti-tamper can be used. Anti-tamper technology can be used to either discover tampering, or to stop it from occurring all together. This system could be placed adjacent (depending on the technology used) of the excising component one wishes to secure. An anti-tamper system is fitted for a pre-defined threat which in turn depends on the security level one wants to achieve for a to-be secured component. Also, the consequence of the tamper is dependent on the anti-tamper technology used.

1.2. State-of-the-Art

Several anti-tamper technologies exists. For the purpose of this research only anti-tamper technologies that prevent a tamper from happening (by acting on a tamper attempt or rendering the system unable

when tampered with). These technologies can then be divided into two categories: active- and passive-anti-tampering. Where active anti-tampering requires a power source to operate and whereas passive methods do not. The integrity of an active anti-tamper system cannot be 100% guaranteed. Since a power outage could allow for a time window, during which the system can be compromised. A passive system that can be fully powered off; at runtime provide integrity checks of the system is more desired.

1.2.1. Active Anti-Tamper Methods

A common high assurance approach is to use a protective foil over (and under) the most critical parts of the design ([11], [12], [13]). And (continuously) monitoring either the resistance or capacitance of said foil. This method is however only on Intergrated Circuit (IC) level. Meaning that it requires to be tightly integrated by the secure system and might therefore not be scalable. A more non invasive method is proposed by Tahoura Mosavirik et al [14] (similar to [15]) by monitoring the entire power network of the PCB. Whenever this changes the circuitry must very rapidly delete any critical information before an attack can extract it. These kinds of anti-tamper measurements suffer from the fact that they need active power in order to function, and must operate extremely fast in the case of an attack. Because if the attacker is faster at extracting the data then the circuitry is to delete it, then all is lost. Deleting FLASH or Random Access Memory (RAM) could simply take too long. And disconnecting Dynamic Random Access Memory (DRAM) from power in the hope that the data will be deleted is naive, since it can take up to 5 minutes for the data to completely be removed at room temperature [16]. It takes even longer at sub zero temperatures, given an attacker ample of time to extract all the data from RAM.

Paul Staat et al [17] developed an anti-tamper system where two antenna's are used to monitor incoming probes. This method provides system wide protection and is less sensitive to quick timing attack's since the attack can be sensed from afar. However, it does require the system to be active at all time. Since the anti-tamper technique is not used to encrypt the data in someway: Simply cutting the power, freezing the system or rapid disconnecting some vital part could still possibly bypass the security.

1.2.2. Passive Anti-Tamper Methods

One of the simplest anti-tamper techniques is simply putting a sticker on the lid, or making use of switches that trigger when opening the box. These are however not sophisticated enough as they do not prevent access once a tamper has occurred. This goal can be reached by having the anti-tamper technology act as a cryptographic key, that gets altered or destroyed upon tampering. The B-TREPID envelop [1] provides a wrapper around the device that has an unique capacitative property that changes when broken. This capacitance is measured on startup and used to decrypt the content of the device if it remains intact. This sort of technique is quite common as a similar to the flexPCB technology represented by Immler et al. [2] where there are multiple points of the foil attached to readout points. These multiple attachment points give the advantage that the value can be used to generate a key. The foil is attached over a single IC (or very small part of the PCB). Where an attacker can only come by destroying the foil and thus also the key. The downside is that the key needs to be reliably extracted or the system will be bricked. This system, however is still vulnerable to rapid key extraction from ram like described above. Furthermore, this foil only protects a single IC (or limited area) on the PCB. Which may in turn lead to greater attack surface. Other work like the research done by Barekatian et al [3] also only cover a single IC. The variation of battery less anti-tamper are not that vibrant and require complex, pricey and limited (area) protection from tampering. The research presented in this thesis will therefore focus on researching a passive tamper detection system, that can ensure the integrity of the whole device (regardless of inside technology and size). And can be constructed without advanced materials or specialized high-tech construction techniques.

1.3. Physical Unclonable Functions

A deployed Anti-tamper system should not be able to get copied. Inherently this is where PUFs come into play. These systems by design cannot be physically copied. Using such a system for an anti-tamper technique gives the system the extra property that learning how to bypass one system does not inherently give way to other instances of the same system as well. Every enrollment of a PUF is different. These variations arise naturally during the manufacturing process of the system and are uncontrollable even by the manufacturer, making each PUF instance physically unique and nearly impossible to replicate [18].

Unclonability: The unique response of a PUF is derived from the random physical characteristics of the hardware. Since these characteristics are determined by uncontrollable manufacturing imperfections, it is infeasible to reproduce the exact same PUF response across multiple devices, even if the design is identical [19].

Challenge-Response Behavior: A PUF operates by responding to a given challenge (input) with a specific output (response). The challenge-response pair (CRP) is used as a form of authentication or key generation, where the response is inherently tied to the physical structure of the hardware. These CRPs can be used in cryptographic protocols to ensure secure authentication or to generate device-specific cryptographic keys [19]. PUFs work by providing a response. Ideally this response is unique for the PUF family and cannot be cloned. And is also the same every time. The latter is practically not the case and thus error correction or so-called helper data is needed.

Classification: PUFs are classified in two categories: *strong* and *weak* PUFs. This concept was introduced first by Guajardo et al. in [20], and further refined by Rührmair et al. [21]. Strong PUFs have a much broader response set then weak PUFs. Therefore, it is infeasible to brute-force the correct response from the device. This means that it has a large challenge set and is its therefore infeasible to build an accurate model for the PUF. Furthermore, a PUF **can** also work by providing a response to a challenge that is outside the environment of the PUF, and could be in the form of a password. This challenge greatly increases the security of the device and (as long as the password input is long enough) is referenced to in literature as a *strong* PUF [22]. Furthermore, by expelling a response once it is given makes a man in the middle attack unlikely. All challenges should produce a different response, and should only be valid for a single time. This makes copying the response harder [22].

Tamper Resistance: Due to the physical nature of PUFs, any attempt to tamper with or reverse-engineer the device typically alters the underlying physical characteristics, thereby changing the PUF's responses. This makes PUFs highly resistant to physical attacks and reverse engineering [19].

Challenges in PUF Design

Although PUFs offer several advantages in hardware security, there are also challenges in their design and implementation:

Reliability: The response of a PUF can vary slightly due to environmental factors such as temperature and voltage fluctuations. Error correction mechanisms are often needed to ensure consistent and reliable PUF responses .

Response Stability: Ensuring that a PUF consistently generates the same response for a given challenge over time is critical for its use in secure applications.

The concept of a PUF has been around for a while and several different implementations exists. Mostly used for unique identification, authentication and verification of a system (like [23], [24], [25], [26]). There exists a wide number of PUF approaches as described by [22]. These differ from silicon PUFs to non-silicon PUFS. Silicon PUFs in nature are not a suitable candidate since they verify the "integrity" inside the IC (like [27]). A PUF that takes the environment as their *uniqueness* is far better suited for anti-tamper and is thus further considered in this paper. These consists of (but are not limited to) [22]:

- Optical
- Paper
- Magnetic
- Radio
- Image

Although optical and image appear the same, they differ in how an optical sensor is used. Where an optical PUF may use a (single or many) photo diode [28], and image PUF on the other hand actually takes pictures at a set resolution and uses image processing to construct PUF data [29].

1.4. Contributions

The goal of this research was to evaluate the feasibility of a passive anti-tamper method. By doing so a number contributions have been made, ranging from experimentation methods to characterize the PUF, algorithm adjustments and software implementations:

- **Glitter PUF**. This thesis introduces the concept of Glitter PUF: A passive (battery-less) imaging PUF anti-tamper method. That uses a camera on the inside of the system to take a picture of its surroundings. This image is used to encrypt/decrypt the contents of the device. The image will change upon tampering, which results in a failure en decrypt the secure data and thus render the data useless.
- **Imaging Particle Reflection PUF Evaluation Metric**. Several characterization methodologies have been introduced to properly identify the behaviour of imaging particle reflection PUFs. These were used to identify the properties of the prototype Glitter PUF, but could also be used to evaluate other prototypes or design iterations. These evaluations metric consist of: finding the most optimal camera configuration, that gives the most information of the enclosure, operational and tamper behaviour identification that is critical to be able to distinguish between them.
- **Fuzzy Picture Difference Algorithm**. In order to succesfully implement Glitter PUF a adaptation of the fuzzy commit algorithm needed to be designed. This modified version takes into account the characteristics of the PUF and the tamper that needs to be detected. Furthermore, by modifying parameters this algorithm also works with different PUFs. The altered algorithm is also 3 times faster to evaluate and uses approximately 3 times less storage of helper data on the device. Making it less prone to reverse engineering by extracting the helper data.
- **Diamond PUF**. An improved version of the glitter PUF. Using (fake) diamonds. Possibly more sensitive to tampering and costing less time to evaluate. Because its more transparent (so it requires less exposure time). And collects more data of the enclosure because it does not block light.
- **Software implementation**: A prototype implementation that demonstrate the feasibility of the algorithm and its ability to detect tampers. This implementation has been written in C++ and is platform independent. Furthermore, the implementation is tailored to give as much information about the tamper and is highly configurable.

1.5. Document Structure

This research document is organized as follows:

- **Section 1** containing the introduction, motivation, state-of-the art and explains PUFs.
- **Section 2** defines the threat model and key extraction techniques used to introduces a new PUF concept for passive anti-tamper, which is further analyzed in this document.
- **Section 3** describes the experimentation for the Glitter PUF defined in section 2.5. And introduces a new improved version of Glitter PUF, that uses diamonds instead of glitters.
- **Section 4** discusses the final implementaion and the usability of the Glitter PUF with the experimentation results gathered from section 3.
- **Section 5** concludes the research by discussing the findings.

2

Passive Anti-Tamper Design

This section describes the contribution of the new anti-tamper design (Glitter PUF). First the threat model in introduced in Section 2.2, along with various key-extraction methods (Section 2.3). Afterward, the concept of Glitter PUF is explained in Section 2.5.

2.1. Introduction

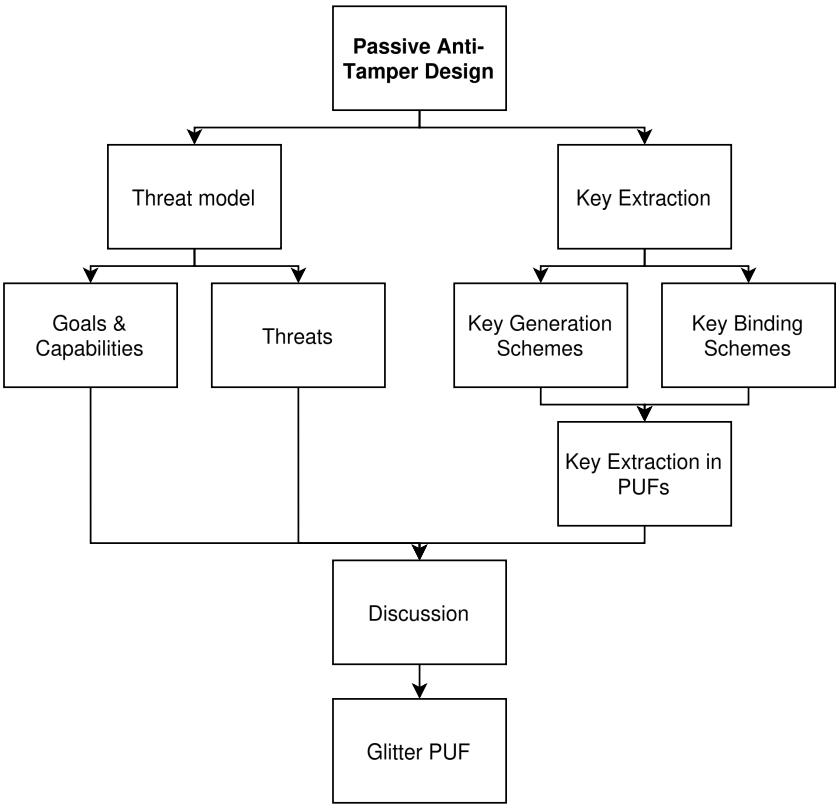


Figure 2.1.1: Chapter overview.

When designing a new anti-tamper method it is important to consider the threat model. So that a solution fitting the requirements of the anti-tamper method can de designed. This threat model consist of goals and capabilities that an adversary may have. Furthermore, several key extraction methods are also considered for encryption. Resulting in an anti-tamper Glitter PUF method introduced at the send

of this section. See Figure 2.1.1 for a complete overview an reading guideline for this chapter.

2.2. Threat Model

Knowing the attack vector[1] is of great importance when characterizing, designing and implementing any anti-tamper method. It is difficult to design a system without knowing what the possible attacks are, why they are important and what methods can be used to mitigate them. Therefore, the properties of the physical device must be categorized. And the most crucial ways to tamper with it, mitigated. The threat model will thereby identify threats and optimize security if mitigated successfully.

2.2.1. Goals & Capabilities

The goals and capabilities describe what an attacker is trying to do to the system (*goals*). And what their means are to achieve that said goal (*capabilities*). Determining these characteristics is important as they will help in designing a secure solution where those goals cannot be reached with the capabilities in mind.

The threat model is focused on physical access to an enclosed device. Where the attacks' goal is to either modify the hardware or extract sensitive data from it.

Attacker Goals:

- **Modify device**: the adversary wishes to alter the functionality of the device, by either adding functionality to monitor, cause a denial-of-service or removing funcionality.
- **Data extraction**: Extract data like cryptographic keys or other vital information stored on the device.

Attackers Capabilities

- Physical access to the device.
- X-ray (or other see-through technology).
- Mechanical knowledge.
- Means to open the device (e.g. by drilling).

2.2.2. Threats

A number of threats exists for the anti-tamper system in question:

- **Spoofing**: The adversary might try to spoof the correct response from the anti-tamper device to trick it that its not been tampered with.
- **Repudiation**: Where the attacker covers their tracks after tampering with the device. So it's not clear that a tamper has occurred.
- **Information Disclosure**: the adversary acquires sensitive data that's on the secured device.
- **Denial of Service (DoS)**: (temporary) disable or disturb (certain) functionalities of the device. These DoS attacks might also happen after a certain amount of time. Making the device suddenly fail when in operation.
- **Elevation of Privilege**: Adversaries could escalate their access privileges within the device.

2.3. Key Extraction

PUFs in general literature are often used for identification purposes and not so much for key extraction or anti-tampering purposes. Biometrics however, almost always are. The sought application may have more in common with a biometric application than classical PUFs. Therefore, a look into how biometric encryption/decryption works is worthwhile.

Different ways of biometric authentication exist. For the sake of this study, crypto systems are further investigated. Crypto systems can further be subdivided into two sub categories: Key-binding and key-generating schemes [30].

[1] methods for an actor to penetrate the system

2.3.1. Key Binding Schemes

Key binding schemes bind a specific key to a biometric dataset. The original *secret* key must be reconstructed by providing a dataset of the original biometric data.

Fuzzy Commitment The original fuzzy commitment proposed by Jeuls et al [31] although fundamental, has some vulnerabilities in it [32]. Improvements to this system have been made [33], [34].

The standard working of the fundamental biometric encryption methods proposed by Jeul et al [31] is done by:

Enrollment:

- Generating random key K
- Encode K with ECC encoder, given C
- Next the original biometric data is XOR'ed with C given AD
- K is hashed ($h(K)$) and stored together with AD in a local database

Authentication:

- AD is being taken out of storage and XOR'ed with newly obtained biometric data, given C^*
- C^* is being decoded using ECC given K^*
- K^* is hashed and compared to the stored hash. If they match authentication is given.

The scheme of fuzzy commitment is pretty straight forward and very flexible. The ECC, hash function and key generation method can be tailored to the specific implementation needs. The problem with Fuzzy commitment is that by storing the hash, ECC and AD, information of S can possibly leak to the adversary. It might not be necessary to store the hash on the device. Since the user will endup with invalid data at decryption. And if the decryption was done succesfully with valid fuzzy data than the device is usable. By just trying to use the decrypted data and see if it functional might be enough.

Fuzzy Vault

Fuzzy Vault works by encrypting secret s with key k so that an alternative key $'k$ can also unlock the so said vault and recover the secret. It does this by selecting features of k. When the alternative key matches enough of these features (that can be weighted [35]) the secret may be revealed. However, in contrast of Fuzzy commitment it does so in an in-ordered fashion [36].

2.3.2. Key Generating Schemes

Key generating schemes use the biometric data as the secret key. So no key generation takes places.

Secure Sketch

A Secure Sketch (SS) consist of two procedures SS and Rec (Sketch and recover respectively) with the following properties [37], [38], [39]:

- The SS procedure takes a binary array as input (w) and returns a binary array output s
- *Correctness*: if $hamming\_distance(w, w') <= t$ then $Rec(w', SS(w)) = w$, thus the original input can be reconstructed.

Fuzzy Extractor Fuzzy extraction is similar to Secure Sketch but does not recover the original input. And requires helper data instead. It uses Gen and Rep (generate and reproduce respectively). The properties are as follows [37], [40], [41]:

- on binary input w and helper data P an output M is generated
- *Correctness*: if $hamming\_distance(w, w') <= t$ where w' is the newly obtained biometric data and t is the threshold, then $Rep(w', P) = R$

No method is perfect and Fuzzy extractor also leaks security information [42].

2.3.3. Key Extraction in PUFs

PUFs can be used to generate keys [43], [44]. Several optical PUFs already use a form of key extraction. Like using lasers [23], CMOS sensor noise [45], [46], or using fabric [24]. These methods for the PUF extraction are also quite variant.

Silk method: For the silk method [24] three different photos are taken using three different coloured lights at three different positions. For error correction the following steps are taken: [24]

- remove noise, by applying a threshold to distinguish peak points
- Binning of the peak points (by means of resizing)
- generate bitstream
- Von Neumann de-biasing process (fuzzy bit extractor)

Another more general approach is discussed by Shariati et al [47]. Here the picture taken is first filtered using binary or Gabor hashing. Next fuzzy commitment is used for error correction and a key is extracted. Mesaritakis et al [23] took the exact same approach for their PUF, but used a different method for the physical part (laser based instead of silk and leds). Their results differ from Shariati's method. Therefore, it can be said that a different PUF implementation highly influences the result of further processing.

2.4. Discussion

The goals, capabilities and threats are kept in mind when designing a proper passive anti-tamper technology. Do note that the research of this thesis is only a proof-of-concept and that further work regarding the threat mitigation capabilities of the model have to be done.

The landscape of key extraction is large with many different techniques that have altering goals and security parameters. These advanced key extraction technologies might distract from the core principles of the anti-tamper technique. Therefore, the most fundamental key extraction technique proposed by Jeal et al [31] (Fuzzy Commitment) is chosen for the initial prototype. Since its form the basis of all key extraction methods that came after it. Moreover, since most key extraction methods are generic and have interchangeable ECC, meaning that the recoverability is the same.

2.5. Glitter PUF

In order to comply to the thread model in section 2.2 an imaging PUF design is picked because it can be used to *observe* the whole system while being technology independent. The key to the system is that any tampering would undoubtedly destroy the visual aspects of the system. In order to not rely on the actual picture of the product (which could easily be reproduced by an attack) the picture must be taken of something absolutely unique per device which will be destroyed upon tampering (e.g. a PUF).

The design is as follows: a wide angle camera is placed in the middle of the secure device, surrounded by LEDS at the base. Then, transparent epoxy mixed with tiny glitter particles are poured over the inside. Filling the entire volume. The system is then closed in a light-tight box. This build procedure is the same for every device. But for every device will result in a different glitter composition, and thus a different picture. Furthermore, the physical glitter particles composition cannot be copied or repeated once settled. By that the concept complies to the PUF characteristics.

Next, the camera can take pictures and use that to encrypt the device. When an adversary attempts to tamper with the device the epoxy and glitter composition will be destroyed and thus the picture will change (see Figure 2.5.1). Note that the exact shape of the enclosure is not taken into account, as a real implementation should completly surround the secure component. But this model is used for evaluation.

The reflection of the glitters is used to encrypt sensative data on the device (using a key extraction method as describes in Section 2.3). Making any change by drilling or sawing open the enclosure will irrevocably alter it. This anti-tamper mechanism works battery-less. Meaning that when the system is powered on the system is de-encrypted using the picture of the glitters as input. Furthermore, this can be done continuously during runtime to brick the device as soon as something is going on. Or even

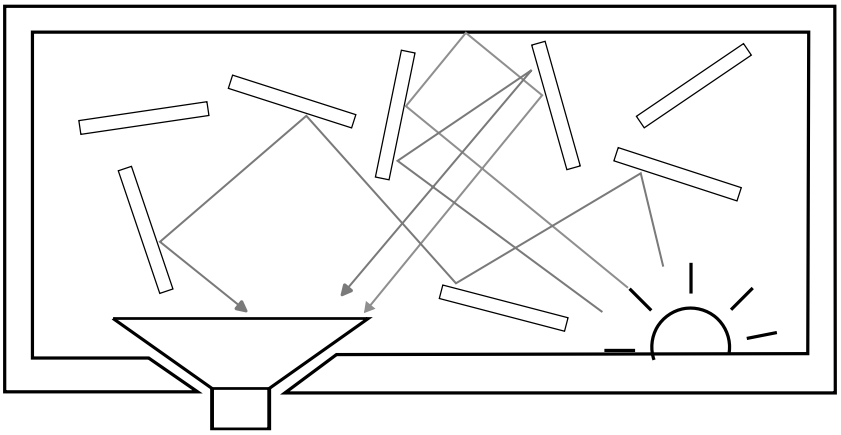


Figure 2.5.1: Sideview of the glitter PUF where the top (above the box) represents the unknown world. And the bottom the protected contents. The cone on the left represent the camera sensor; the ball on the right the LEDs; the rectangles the glitter particles.

when the device is turned off and back on again while there was tamper in the meantime. This will also be detected.

Furthermore, an extra challenge can be used in the form of a password so only authenticated users can access the device (making it into a strong PUF). Glitters form a good basis for such a system as it will reflect light. In an ideal scenario, all the pixels of the image should be influence-able by every part of the inside of the enclosure. So that when an evil operator drills inside the encasing the image should change. Even if the drill is not in the Field-of-View (FoV) of the camera. By using glitters the goal is to make the light reflect inside the enclosure as much as possible and making sure that each pixel gathers as much information about the enclosure as it can. See the side-view diagram of the Glitter PUF in Figure 2.5.1.

The concept of glitter PUF is highly scalable as the resin solution can be poured over any material (being a PCB, Hard Disk Drive (HDD) or any other technology that stores data). As long as the objective is to protect the underlying data securely. Furthermore, the concept also works transparently and does not require any interaction from the end-user. The device will work if not tampered with, or it will fail to boot (if the firmware in encrypted using the glitter picture). A method like this will prevent an adversary to extract *secret* data on the product. And prevents the product from functioning if any modification attempt have been made. Thus mitigating the attackers goals. Furthermore, this concept is cheap. As the only hardware requirements are a camera, resin and glitters.

3

Characterization

This section describes to characterization of the Glitter PUF. First the metrics and test setup is described (Section 3.3 and 3.2) followed by the performed tests. The best picture settings are first determined (Section 3.4) followed by temperature and aging-tests (Section 3.5 and 3.6). This is used for for comparision agains tampering, done in Section 3.7. Afterwards a new PUF design is decribed in Section 3.8

3.1. Introduction

In order to assess the feasibility of the Glitter PUF it is nessesary to first devine metrics for characterization. These metrics are used in all further tests. Then the best picture settings are sought after. Finding the best picture settings is important as an unprocessed image with as much information is needed. These picture settings are used in the characterization tests. Which are divided into two categories: functional tests and tamper test. Functional test, like the ageing test are used to characterize the system when in operation, while the drill test is used to see how the system reacts to tampering. A drill test is chosen because small punctuations are the biggest threat to the system.

To evaluate the feasibility of the Glitter PUF different forms of experimentation are considered:

- **Best picture settings**: different picture settings like exposure, shutter speed and amount of light exposed to the CMOS sensor yield different picture results. The goal of the image is to have as much information of the environment as possible and that parts of the image are not biased, or processed by software. Ideally the most bare and unprocessed result containing as much information is desired.
- **Picture difference by temperature**: Temperature can alter the formation of the glitters and resin. Furthermore, the camera sensor used is not immune to temperature differences. This could create a problem when trying to validate the image when temperature-cycles have been taken place.
- **Picture difference from drilling**: The main form of alternation from an adversary is by drilling in the enclosure (as stated by the thread model, Section 2.2). These test are used to give an insight of how much protection the glitter PUF gives.

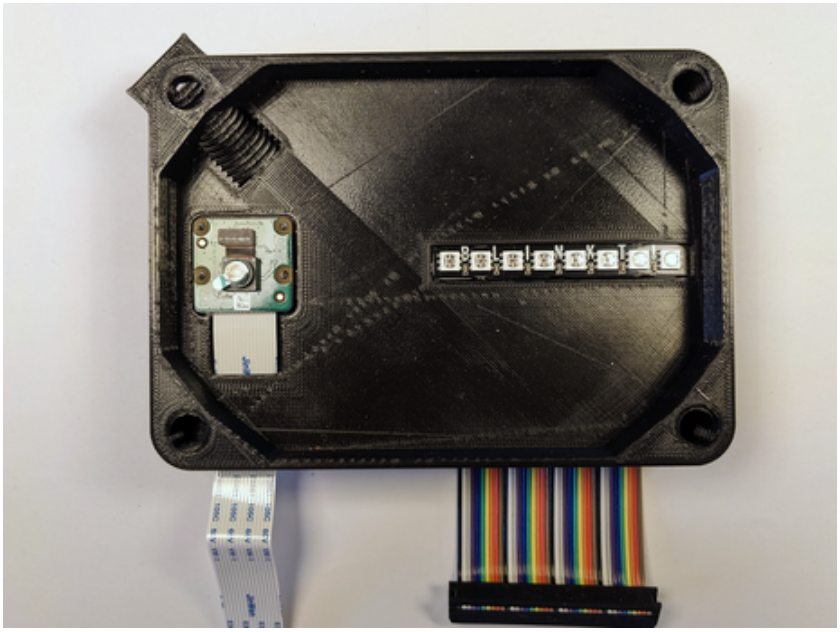
3.2. Test Setup

For the purpose of the experiments the following hardware configuration is used:

- RPi 4
- RPi Camera Module 2 (Sony IMX219)
- 3D printed case
- 8 Red Green Blue (RGB) ledstrip (5 LEDs were used)

The enclosure is 3D printed using Poly(lactic acid) (PLA) and is 12mm tall, 100mm wide and 140mm long. The led strip is positioned perpendicular next to camera at a distance of 35mm (see figure 3.2.1).

Note that the characterization is that of the feasibility of the concept. And that the shape of the box and camera, LED placement is not taken into account. In a real implementation the glitter PUF should completely surround the device.



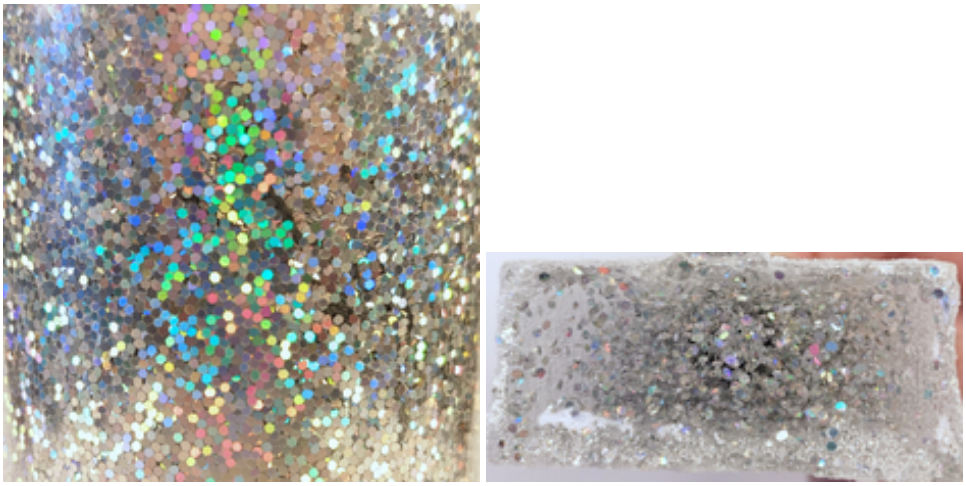
(a) Front view of the PUF enclosure when opened, without epoxy

(b) Front view of the PUF enclosure when closed, filled with epoxy. Note the drilling hole on the left center (further discussed in Section 3.7).

Figure 3.2.1: Realization of the PUF used for characterization.

The glitters (Figure 3.2.2) are mixed with the UR5634 polyurethane resin from Electrolube. The resin is developed to encapsulate electronics, can operate from -50°C to 150°C and is transparent [48].

The camera is operated in raw mode, for more information see Section 3.4.1 and Appendx A.



(a) Glitters, used in tests. **(b)** Glitters inside transparent PUR, used in tests.

Figure 3.2.2: The glitters (inside PUR) used for testing.

3.3. Metrics

Characterization requires metrics in order to successfully compare results and make informed decisions based on the acquired data.

Shannon Entropy [49]: Shannon Enropy determines the randomness in the data; is used to determine the amount of (unique) information a set of data contains.

Concatinated Colour Distance: Comparing images to eacher is a multidimensional problem. As an images can have the same values in one location and differ from another spot. Furthermore, a pixel is made out of three components (RGB). These colour channels must be compared individuality to another. Therefore, in order to create a scalar metric the concated colour distance of all the channels (e.t. sub-pixel) for each pixel in compared to another image. Mathmaticly the formula looks like this:

$$d = \sum_{n=0}^{n=s} \sum_{m=0}^{m=2} |a_{nm} - b_{nm}|$$

where, a and b are different pictures, s is the total amount of pixels (requiring to be the same for both a and b).

This colour distance is used to compare images, and to see how much different they are. Furthermore, this metric is chosen above signal-to-noise ratio or Root mean square (RMS) because the colour distance gives a direct insight of the recoverability and ECC requirements. As an ECC can be designed to correct a certain threshold of colour distance.

Structural Simularity for a PUF it is desired that all the response's (given the same challenge) provide the same output. However, the glitter and resin competition might fluxuate and differ. It is therefore required that to measure how much two images are alike, without also taking noise sources like the Complementary metal oxide semiconductor (CMOS) imaging sensors into account. Structural Similarity Index Measure (SSIM) is defined by Wang et al. [50] and is paralytically useful to determine how similar two images are to the human eye, regardless of small changes in contrast, histogram or noise.

3.4. Picture Settings

In order to evaluate the Glitter PUF further it is necessary to determine the best picture settings to ensure that every detail is captured. In order to determine this the Shannon entropy [49] is used as a metric. Since this shows the variety of information within the picture. The higher the score, the more chaos and thus possible information exists inside the picture. Furthermore, to determine if there is actually a significant difference between measurements, the colour distance (for each colour channel (RGB) i.e called sub-pixel) is taken into account. This colour distance is the absolute difference between the reference and the test pixel colour. Note that this distance is concatenated for each pixel for each channel. And gives thus a single scalar output for each comparison.

3.4.1. Photo Configuration

Getting a picture that show every detail (thus having as much information of *that* what is being pictured) is of vital importance to this application. A proper configuration of oversampling (to remove noise), camera settings, shutter-speed and exposed light is therefore necessary. As these elements all influence the resulting picture.

Camera Settings

Camera configuration is a big part of photography. Many parameters can be adjusted such as International Organization for Standardization; ISO 5800:2001 (ISO)[1], shutter-speed, exposure time, diaphragm, gain and focus point etc. It is however, desired to have an raw and thus unprocessed image. That has the same parameters for every picture taken. As post-processing, ofter done in software will alter the image. These alteration can introduce noise (ISO and gain), and possibly remove information (e.g. if exposure time is set to high or to low).

So all pictures have been taken using raw settings, with minimal gain and ISO in order to get an as unprocessed response from the image sensor with static parameters for colour balance and focus point (see appendix A).

Samples

Every sensor is subject to noise. As is the imageing sensor used in this prototype. This noise is undesired as it could conflict when comparing images. As describes in Section 3.3 comparing image based on their colour differnce is desired. Inorder to properly achieve this, noise must be minimized.

A way to minmize noise is to over-sample the image. The average value of all the samples is then calculated and used for further processing. The optimal number of oversampling must however be determined. Figure 3.4.1 shows the entropy (normalized) agains the amount of samples used to calculate the average value. Shannon Entropy is used as a metric because for this use-case less entropy means less noise. Looking at Figure 3.4.1 it can be seen that the noise reduces inverted logarithmically. Moreover, looking at the graph it can be said that 20 times oversampling has diminishing returns (as the entropy in the image flattens in Figure 3.4.1). An over-sampling of 6 might also be sufficient. But for the purpose of this research it is chosen to over-sample by 20 in-order to have as less noise as possible. But for production purposes 6 might suffice.

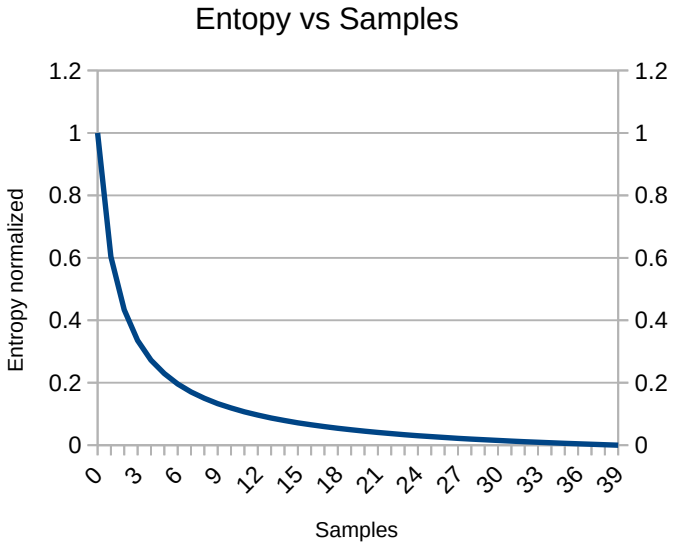


Figure 3.4.1: Shannon Entropy (normalized) agains samples taken

Shutter-speed and Light

Shutter-speed and LED configuration determine the amount of light that is picked-up by the camera

[1] describes light sensitivity

LED	shutter speed (s)		
1	45	50	55
3, 1	45	50	55
4, 3, 1	40	45	50
4,3,2,1	40	45	50
4,3,2,1,0	20	22	24

Table 3.4.1: Best Shutter speed and LED configurations used in further testing.

sensor. Ideally this must not over-saturate or undersaturate the sensor, and be reproducible. Experimentation has been done to determine the optimal shutter-speed and LED configuration showing the most information in the resulting image and be reproducible.

Shannon Entropy is weighted with SSIM in order to create a score showing how stable and how much information a configuration set contains. SSIM is used to compare the average of five samples per setting to 3 other samples of the same setting. Resulting in a similarity score. The Shannon Entropy is calculated on the average image per configuration (using all 20 samples).

These measurements are performed with 5 different LEDs in different combinations (totaling at 31) inside the casing (at increasing distance from the camera) with 21 different shutter-speeds, this amounts for 651 different combinations. See Appendix B for more information about the setup.

The results of the best picture settings can be seen in Table B.0.1. It can be seen that the most entropy is achieved with a shutter speed between 40 and 55 seconds for most configurations. Especially when only a few lights are used this setting is most optimal. With the exception when all lights are on. Then the optimal shutter time is between 20 and 24 seconds.

Table B.0.1 shows the weighted score of each configuration. From this the best configuration per amount of LEDs that are on is chosen together with the neighboring shutter-speed. The result can be seen in Table 3.4.1. These configurations are used in all further experiments in this research.

The long shutter-speed of approximately 50 seconds is not desired in a production application, especially when oversampling. As waiting for so long to detect a tamper is tedious. This is however not an issue for the proof-of-concept. As the long exposure time is possibly though the glitter particles blocking much light. This could be mitigated by using smaller transparent particles.

3.4.2. Enhance

Normal pictures can hide information in low contrast parts that may be important for tamper detection. This is not desired as these low contrast part might show vital information of the image. As should have as much weight at comparison. To achieve this, it is necessary to have some sort of contrast equalizer, so all details have as much weight. For this research contrast stretching, histogram equalizing, adaptive histogram equalizing and local histogram equalizing are taken into account (see Figure 3.4.2 for examples). Because it are common equalizing techniques.

Contrast Stretching will keep the area of the histogram the same. But will stretch it out over the entire histogram range. The overal area remains the same by decreasing the intensity.

Histogram Equalizing also known as normalization works by expanding the histogram (linearly) to the entire range (0-255).

Adaptive Histogram Equalizing computes multiple histograms of several parts of the image. And uses those to redistribute the lightness.

Local Histogram Equalizing although similar to Adaptive Histogram equalizing, however it also takes the local neighbouring pixel values into account. So that small details, get emphasized more.

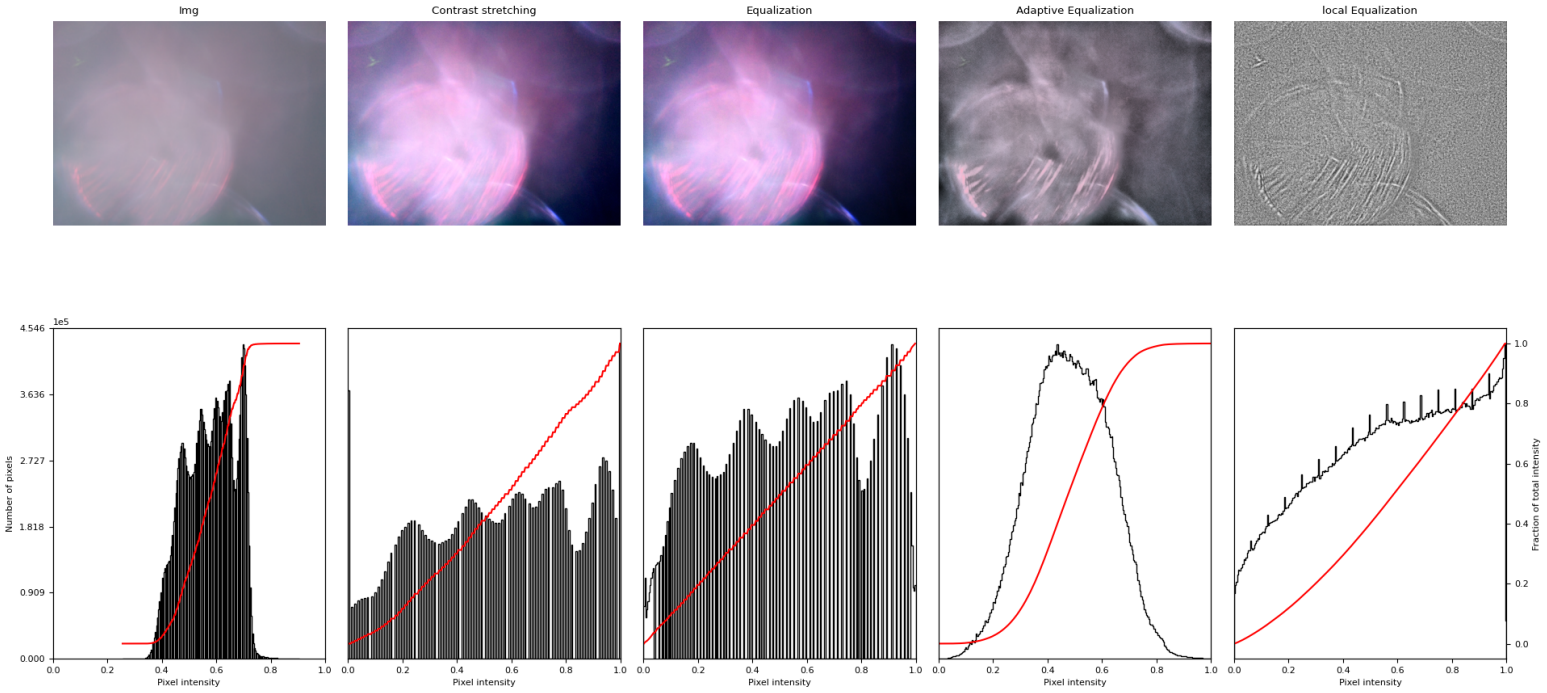


Figure 3.4.2: Reference picture with respected histogram on the left. Followed by several alternations to increase overal details in the image (with their respected histogram output.)

Figure 3.4.2 shows the histogram of the different equalizing techniques. Equalizing simply stretches the histogram to the full range, altering the colours in the process. Contrast stretching does so as well. But seems to lower the values by about half. Furthermore, on these two pictures it can be clearly seen that binning (of the histogram) took place. The Shannon entropy of the enhancement is as follows:

- Default: 6.51
- Constrast Stretching: 6.4
- Equalization: 6.51
- Adaptive Equalization: 19.46
- Local Equalization: 7.91

Adaptive Equalization scores higher than any other enhancement function. And is therefore a suitable candidate. This is revisited in Section 4. This enhancement could however also alter the image in unsuitable ways, and bias other measurements. Therefore, when doing other experiments the default, without any equalization, is still used in order to get the most unprocessed, unbiased image as possible.

3.5. Temperature Test

The temperature test is used to determine the differences in picture settings under different temperature conditions. The test is done in the range of -10°C to 60°C (with intervals of 3°C) because it is a common operation temperature for equipment. The test starts from 22°C goes down to -10°C, up to 60°C and then down to 22°C. Given a full period of testing, therefore (almost) every temperature is tested twice in this full loop temperature test.

For every temperature and setting: 20 pictures have been taken which are averaged into one picture to filter out the noise. For each of these *mean* pictures the total colour distance (for each subpixel) to the reference has been calculated (see Figure 3.5.1).

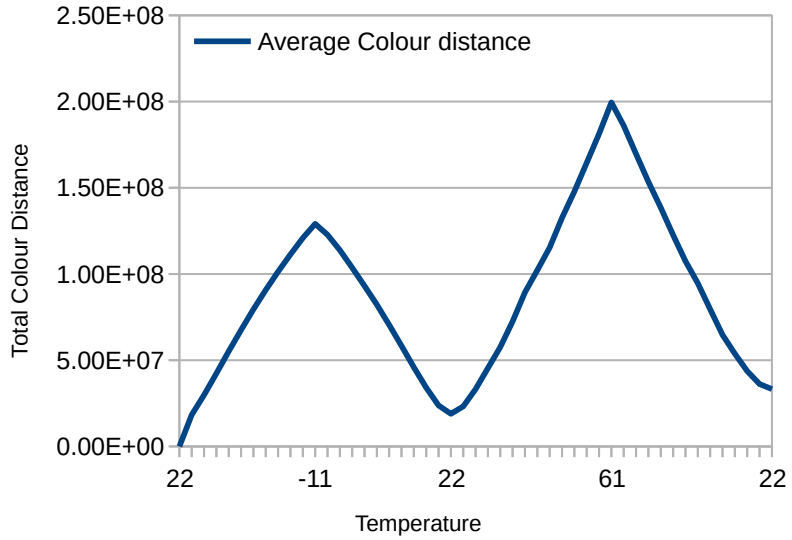


Figure 3.5.1: Total average colour distance (of configurations in Table 3.4.1) to the reference for each sub-pixel for all pixels. When going through a temperature cycle

The first measurement is done at 22°C. The total colour distance is compared in Figure 3.5.1. Here it can be seen that the colour distance goes up when cooling. Then down when the device is 22°C again and shoots up when going to 60°C. The colour distance however, never fully goes back to 0 (even at 22°C at later stages).

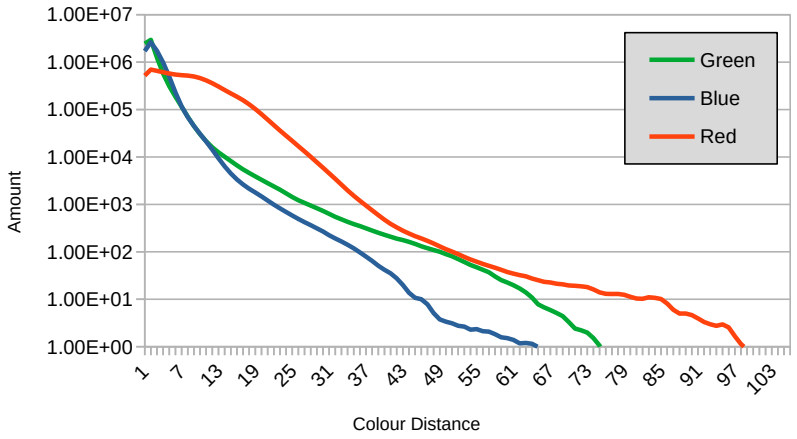


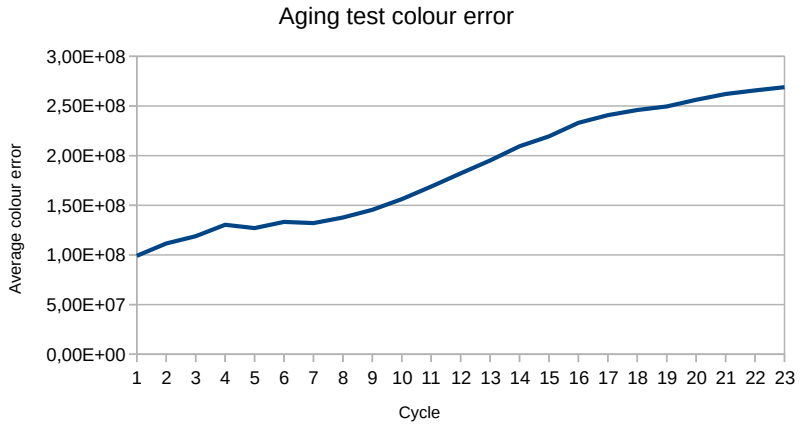
Figure 3.5.2: Average (of configurations in Table 3.4.1) colour differences for the temperature cycle separated on the RGB channels.

Figure 3.5.2 shows how many times a certain colour distance is reached (during the whole temperature test); separated in RGB colour channels. It can be clearly seen that the channels have different stabilities. And that the red channel is clearly lacking. This is important information as discussed in Section 4.3.1 this can be used to filter unstable data.

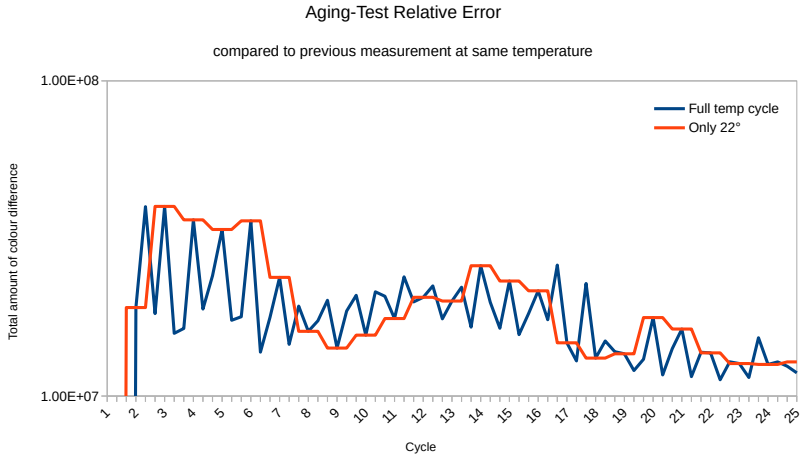
This increase in noise could be problematic if it does not converge at some point. So in order to better understand the behaviour of the system in time an ageing-test (which consists of multiple temperature cycles) is introduced.

3.6. Ageing Test

The ageing test was designed to simulate ageing and to figure out how pictures could differ over time. This has been done over 23 temperature iterations. Where a measurement for each iteration the temperature is cycled through: 22°C, -10°and 60°C depicted in Figure 3.6.1. On this figure it can be seen that the average error is flatting out, and not increasing linearly. Therefore, it can be said that first baking the PUF for approximately more than 20 iterations will harden and stabilize it.



(a) Average absolute colour offset for all channels using configuration of Tab. 3.4.1.



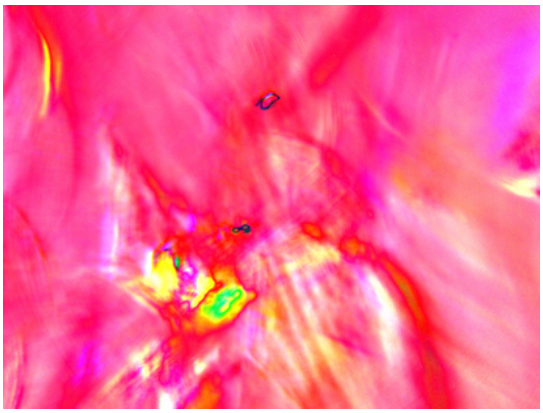
(b) Average relative colour distance based on the previous temperature cycle.

Figure 3.6.1: Plots showing the result of the colour distance introduced in the ageing-test when going through temperature iterations.

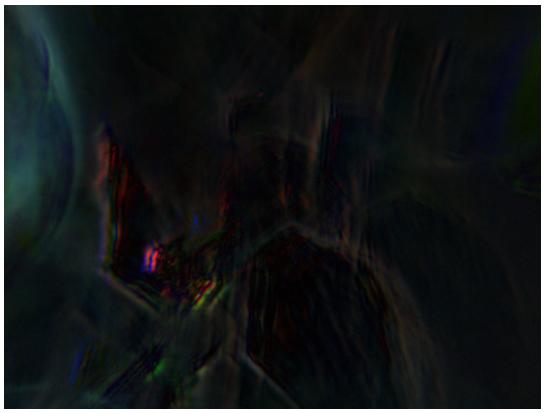
Differences on the (mean) iterations can be seen in Figure 3.6.2. Figure 3.6.2a shows the absolute maximum difference after 23 temperature iterations. This information is extremely useful as these differences show the unstable (sub)pixels of the PUF. Like discussed in [22] a stable or "golden" sample of the hardware must be taken in order to get a stable consistent response from the PUF. For the glitter PUF these golden (sub) pixels can be determined by going through several temperature cycles (20+) and checking which (sub)pixels change the least. Furthermore, this will also bake and harden the epoxy. Making sure that it settles before enrolment. Figure 3.6.2b shows just how small the difference is between two iterations at the same temperature. At the same time Figure 3.6.2c shows that when changing the temperature the difference is much higher. However, this is most noticeable in the red colour channel. These unstable (sub)pixel might be able to be filtered out.

(a) Colour difference of the first iteration at 22°C compared to the 23th iteration at 22°C

(b) Colour difference of the 22th iteration at 22°C compared to the 23th iteration at 22°C



(c) Colour difference of the 22th iteration at 60°C compared to the 23th iteration at 22°C



(d) Colour difference of the 22th iteration at 22°C compared to the 26th iteration at 22°C

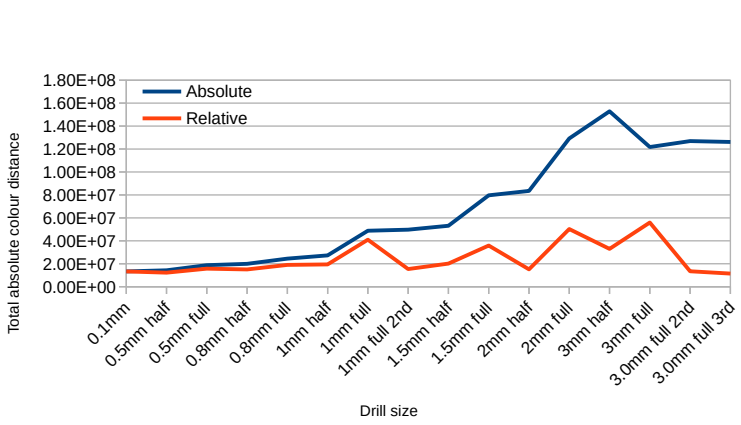
Figure 3.6.2: Pixel differences between age test cycles at various degrees and cycles (with a scale factor of 25).

3.7. Drill Test

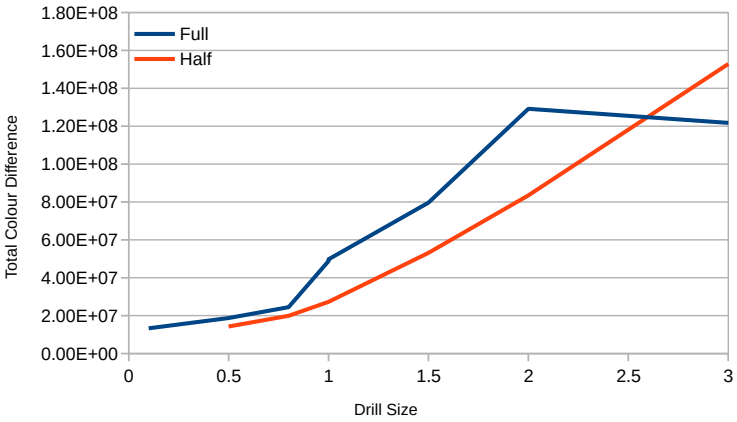
The biggest threat on the system is when a perpetrator drills in the enclosure and manages to alter the system without being detected. All the experimentation beforehand was to categorize the system and know its characteristic under certain conditions at rest. Leading upto the penetration experiment to determine if a tamper can actually be reliably detected or not.

Penetrations are done with drills of diameters: 0.1mm, 0.5mm, 0.8mm, 1mm, 1.5mm, 2mm and 3mm in ascending order. With pre-drilling of a 3mm drill in order to remove the outer shell (and make space for the drills). All holes, with the exception of the 0.1mm since its not long enough, are drilled in two steps. With pictures taken in between and compared to the reference (in this case being the 3mm pre-drill).

The initial results of the drilling show that the total colour difference increases when the PUF is punctured. Especially for bigger holes like 1mm, show big differences compared to the baseline (see Figure 3.7.1a). Different lightning and shutter-speed settings do not make much of a difference.



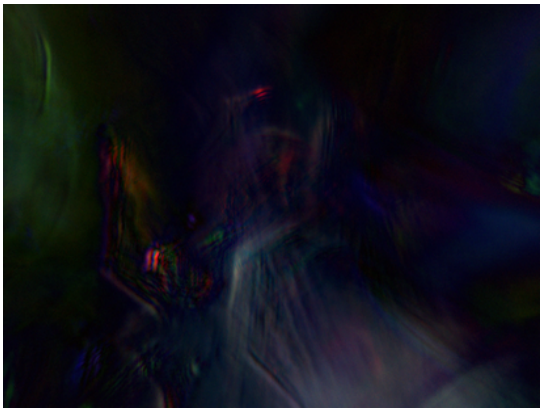
(a) Total colour distance to reference when drilling. Taking the average of all the settings.



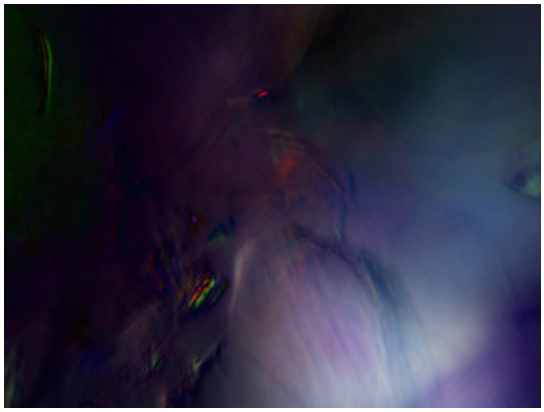
(b) Total colour difference of all drill tests. Only showing absolute difference

Figure 3.7.1: Total pixel error from the drilling tests.

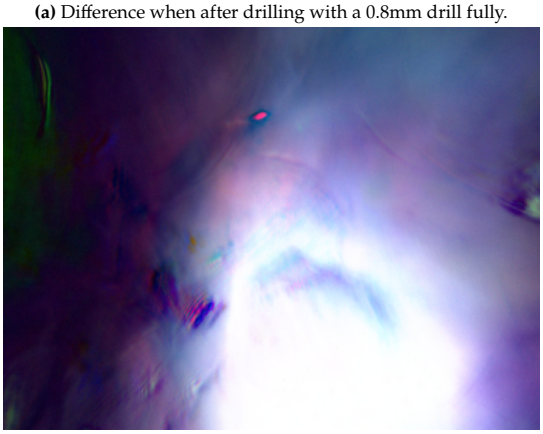
Ideally the colour distance of the drilled hole should be bigger then the relative error of the stable outcome of the ageing-test (Section 3.6, Figure 3.6.1a) which occurs at around the 1mm drill. These one dimensional tests may be misleading as they do not account for the actual location of the error. Nor how each pixel is influenced.



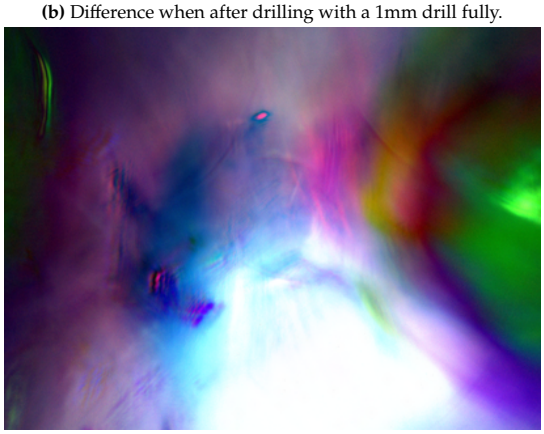
(a) Difference when after drilling with a 0.8mm drill fully.



(b) Difference when after drilling with a 1mm drill fully.



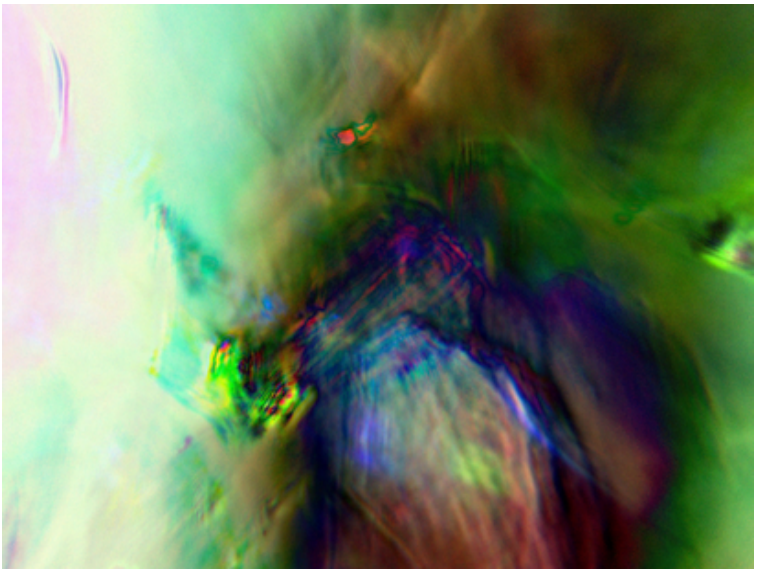
(c) Difference when after drilling with a 2mm drill fully.



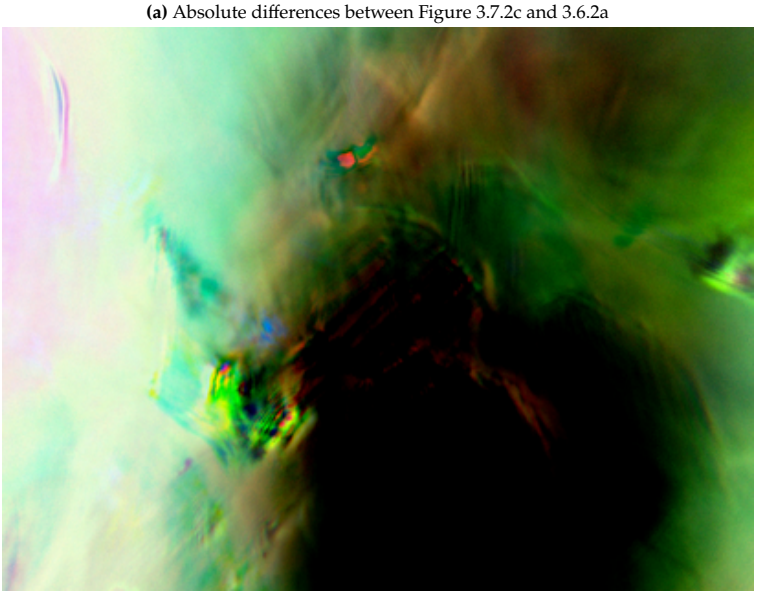
(d) Difference when after drilling with a 3mm drill fully.

Figure 3.7.2: Differences (amplified 25 times) in the drill test of the mean of the picture settings. For a complete set of all drilling differences pictures, see Appendix D.0.1

Figure 3.7.2 shows the difference between the reference and the picture taken after the hole was drilled (absolute difference amplified 25 times). The figure shows a clear increase of differences in the bottom right as the drill size grows. While the left side of the image remains unchanged. This is most noticeable on Figure 3.7.2c. This difference appears white, thus meaning that all three colours differ. This is different from temperature difference (like Figure 3.6.2 and **??**) where mostly only a single channel on a pixel fails. This positional change of all three channels can be used to implement a proper error correcting scheme (discussed further in Section 4.3). An interesting observation takes place when putting Figure 3.7.2c next to 3.6.2a as the active pixels seems to be approximately inverted (see figure 3.7.3)! This is useful information as discussed in (Section 4.3.1) unstable (sub)pixels can be filtered out. And coincidently the pixels changing when drilling are stable pixels. This will make tamper detection much easier.



(a) Absolute differences between Figure 3.7.2c and 3.6.2a



(b) Non-Absolute differences between Figure 3.7.2c and 3.6.2a

Figure 3.7.3: Differences between 3.7.2c and 3.6.2a, clearly showing a dark stable spot after the temperate testing which is obscured when drilling. See Appendix F for further details and additional figures.

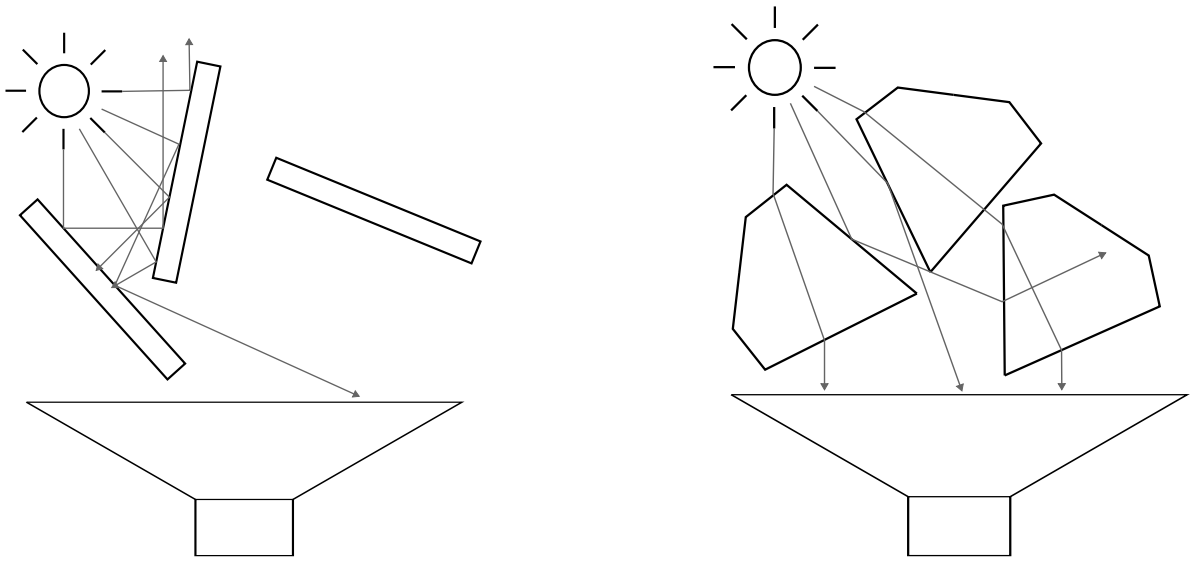
3.8. PUF Design Improvements

Characterization done in this chapter show that an extreme long shutter time is needed in order to aquare the most information in the picture. For prototyping and assesing the concept this is fine, but in a real word scenario this might take to long. This is possibly because of the glitter particles used in this prototype. Futhermore, the camera position, resin used and enclosure shape might not be optimal. In this subsection a new concept called diamond PUF is introduced. That uses a different camera, LED placement, enslosure shape and reflective particles.

3.8.1. Concept

There are many different aspect one can change of the PUF. Including but not limited to: filament, enclosure shape, enclosure material, camera placement, type of camera, light placement, type of light, glitter amount, type of glitter etc. This makes creating the optimal PUF an extremely hard and complex exercise. For the purpose of this research only the type of glitter is considered. Because this can possibly be the biggest factor. The glitters used in the previous PUF implementation are not see-through. This

heavily limits the information that can ultimately reach the camera. A more optimal scattering object would be a prism like object. This will break the light into a set direction and make concatenating "glitter" objects have added value since they do not remove information (see Figure 3.8.1b).



(a) Rectangles represent glitter fragments. Because glitters are not transparent much of the light information gets reflected away from the camera and is thereby lost.

(b) Diamond like figures break the light into different paths, but do not block light and thereby not obstruct the information.

Figure 3.8.1: Diagram (sideview) showing the difference in light propagation. Light source on the top right, and the camera in the bottom as a cone.

shape: To properly evaluate the PUF an enclosure where every part has an equal distance from the camera is needed. This will ensure that the location of the intrusion is known and that only the distance to the camera and the angle is evaluated. And that the enclosure with any dead corners or weird shapes are not taken into consideration. This design is however only for evaluation purposes. On a real implementation a smaller and more practical enclosure might be considered. See Figure 3.8.2 for a side-view of the evaluation model.

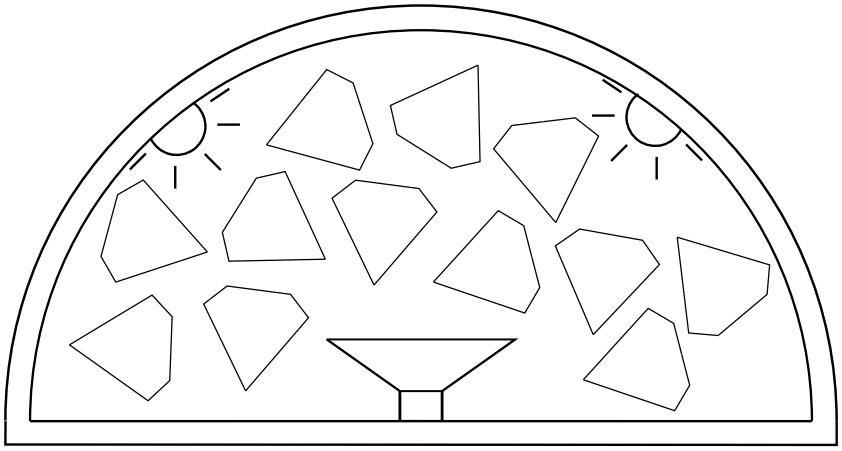


Figure 3.8.2: side-view of PUF design. The construction is that of a sphere with a light strip at a 45°angle (going around the camera)

3.8.2. Realization

Several fake (glass) diamonds are taken in consideration in-order to know which reflects the most information into the camera sensor. The diamonds have different shapes and sizes. One of the diamonds

are made of (partly) iridescence[2] material. See Table 3.8.1 for a complete list of the materials and figure 3.8.3 for reference pictures.

Diameter (mm)	Material	Shape
10	glass	diamond
8	glass	truncated icosahedron[3]
6	glass	truncated icosahedron
4	glass	truncated icosahedron
6	iridescence acryl	diamond

Table 3.8.1: Different materials taken into consideration as replacement for glitter.

(a) Collection of 6mm in diameter iridescence acrylic diamonds.

(b) Collection of 8mm in diameter glass truncated icosahedrons.

(c) Collection of 10mm in diameter glass diamonds.

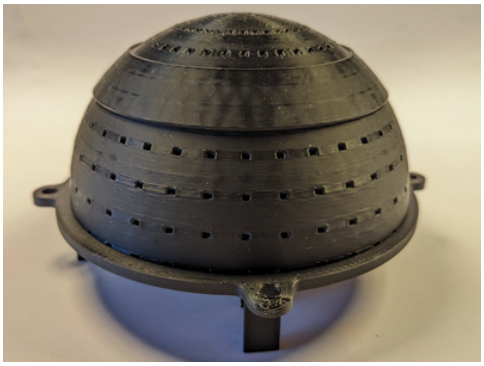
Figure 3.8.3: A sample of the diamonds considered for the diamond PUF design.

The PUF as described like above has been build using a Raspbarry pi 4, RPi HQ camera (Sony IMX477), 180° lens, PLA filement, and a RGB ledstrip. For quick protoying two versions have been made. One with 100mm and a smaller version of 60mm. This is done so that on elementry test not a lot of resin and diamonds is required and the smaller version can be produced faster.

Unfortunately because of timing constrains and scoping of the research this prototype has not been further evaluated.

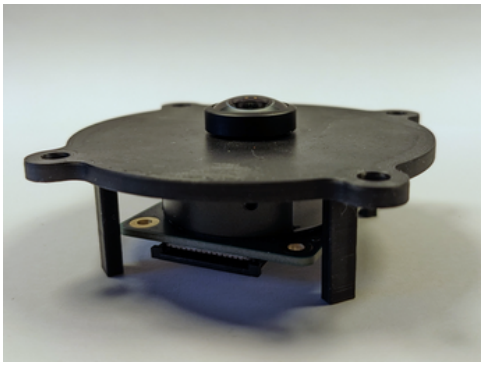
[2]also know as goniochromism, reflects different colour based on the angle of the observer
[3]shape like a football

(a) Big Diamond PUF: 180° camera lens placed into the open enclosure. With the camera sensor underneath.

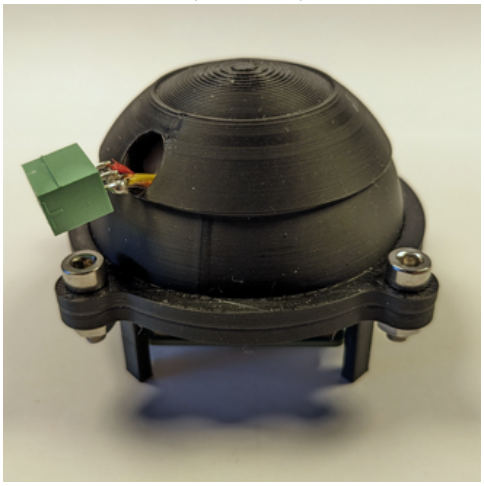


(b) Big Diamond PUF: Cap closed, consealing the device. Currenty without any resin

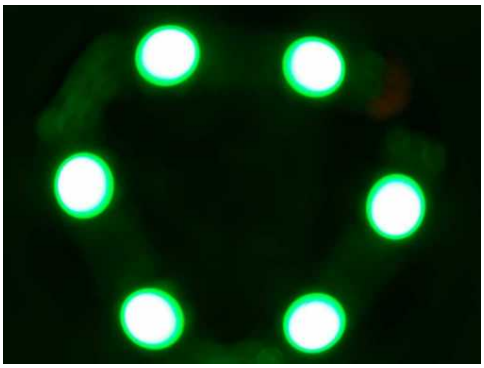
(c) Small Diamond PUF: Inside view of the cap with a circle of RGB leds.



(d) Small Diamond PUF:180° camera lens placed into the open enclosure. With the camera sensor underneath.



(e) Small Diamond PUF: Cap closed, consealing the device. Currenty without any resin



(f) Small Diamond PUF: Picture taken with the camera from the RPi. With leds turned on.

Figure 3.8.4: Pictures showing the prototype's of the diamond PUF enclosure.

4

Implementation & Results

This section describes the implementation of the anti-tamper software. Using the characteristics from Section 3. Segmentation of the image is needed as described in Section 4.2. Different types of error correction are describes in Section 4.3, following by enhancement (as described in Section 3.4.2) in Section 4.4 and greyscaling of the image in Section 4.5. In order to filter and compress the data needed for processing. Afterwards, the implementation flow is explained in Section 4.6. The implementation is evaluated using the drill pictures (from Section 3.7) and ageing pictures (Section 3.6) in Section 4.7 and 4.8.

4.1. Introduction

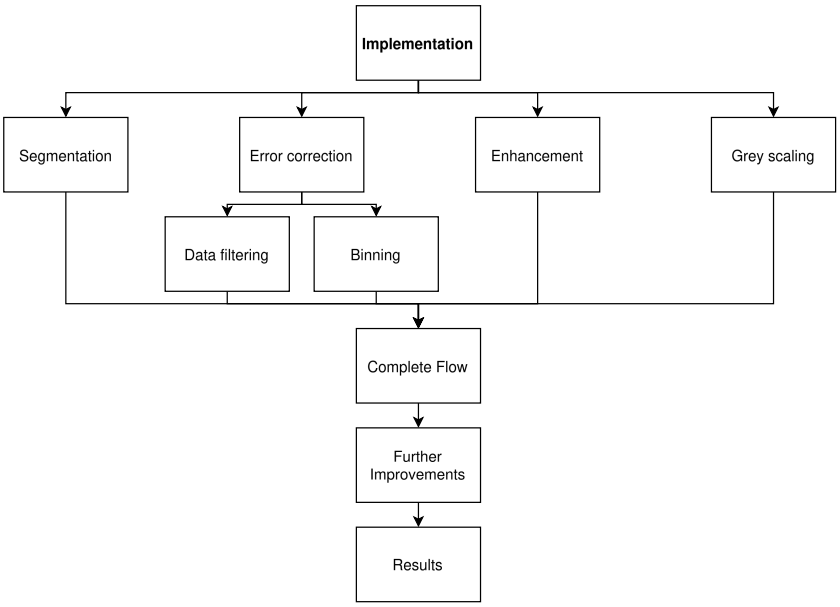


Figure 4.1.1: Reading guideline of this chapter.

Previous charaterizations are used in order to design a suitable software implementation and using a key extraction scheme from Section 2.3. The goal of this software implementation is to make a proof-of-concept; to assess the feasibility of the Glitter PUF. Using the pictures gathered from Section 3 certain design decisions are made. In order to distinguish between functional operation and tampering. The image gathered from the PUF is segmented, error corrected and enhanced. Furthermore, compression using greyscaling is also done. The complete flow of this chapter can be seen in Figure 4.1.1.

4.2. Segmentation

Fuzzy -commitment, -vault and -extraction work on the entire subset of data. The algorithms allow for a configurable maximum hamming distance to the original message for the secret to still be exactable (see Section 2.3). The error however does not need to be uniformly divided over the data. This means that a big error in one part of the image is weighted as much as a small error spread out over the entire image. This is not desirable for drill detection. As can be seen in Figure 3.7.2 that drilling causes a big error in one spot, while random noise through time and temperature deviation occur spread out over the image (Figure 3.6.2c).

The image can be subdivided into a grid where each part of the grid holds a part of the secret key. Then when one part fails because of a drill the key cannot be reconstructed. However, if the other parts of the grid fall within the error margin then the key can still be partially constructed. Therefore, it is necessary to contain information for the entire key into each grid space, or obstruct it in some way so that it is not clear which bits are actually missing. The obstruction can be resolved by using a passphrase upon unlocking. Perhaps the most secure method is to XOR the grids together to get a new value.

In the picture a total of approximately 67349 segments ($(3280 * 2464 * 0.2)/24$) are present, each with a 24 bit random value. These random values can be further used to generate a proper length secret key (like 256 bit for Advanced Encryption Standard (AES) for example). Instead of 24 bits, 256 could also been chosen for segment size. But 24 bits is a good size for segmentation since it lowers the change of left over bits (since it's in the table of 8) and is small enough so it can be quickly calculated and gives an accurate insight on the recoverability of an image which is the goal of this research. Whereas 256 bit might be significantly slower, and might not be nicely segment-able. A way to create a secret key from these segments could be to concat segments until the desired key length is reached. Then *xoring* that with all other concated segments of the same length. Or concating all keys and calculating the 256 Bit, Secure Hash Algorithm (SHA-256) sum over them.

Since the goal of this implementation is a proof-of-concept a segment size of 24 bits is chosen.

4.3. Error Correction

A big challenge when creating an anti-tamper system is to create a system that triggers on true positives and that false positives do not occur. However, the picture already changes to the original inevitably when changing the temperature (as can be seen in Section 3.5 and 3.6). Futhermore, as can be seen in Section3.4.1 the image gathered from the PUF is susceptible to noise. A correct error correction scheme is vital to a correct implementation.

Several approaches to error correction have been considered. Ideally an ECC that fits the characteristics of the PUF most closely should be used. Since that will result in the most errors fixed, but it should also still be secure and not leaking any information about the reference image.

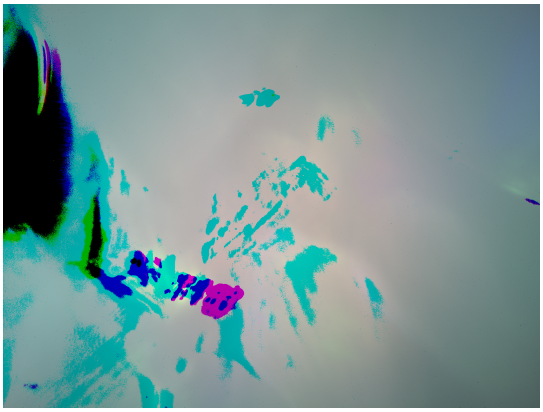
4.3.1. Data Filtering Techniques

Some parts of the image (as can be seen in Figures 3.5.2 and 3.6.2) are more susceptible to noise than others. Filtering these unstable (sub)pixels from the stable ones can greatly enhance the usability of the data. However, one must be careful since interesting parts (e.g. that change drastically when being tampered with) could also be filtered away. Two solution were considered:

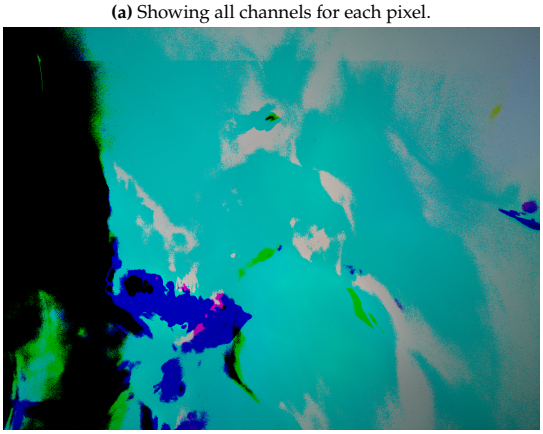
Max colour difference masking:
Masking the image by only allowing (sub)pixels with a maximum absolute colour distance of (for example) *0, 3, 5, 10, 20, 40, 80 or 160* from the baseline. By doing this the maximum error of the whole picture is known.

Best (sub)pixels masking:
The total error of the picture can be brought down by only allowing the best (sub)pixels for authentication. Determining the best (sub)pixels can be tricky. One method could be to let the device go through numerous temperature cycles and then check which (sub)pixels change the less. Dividing the image to sub pixel level (i.e. Red, Green and Blue) is important as different colour channels may have different stability. Example of this can be seen in Figure 4.3.1.

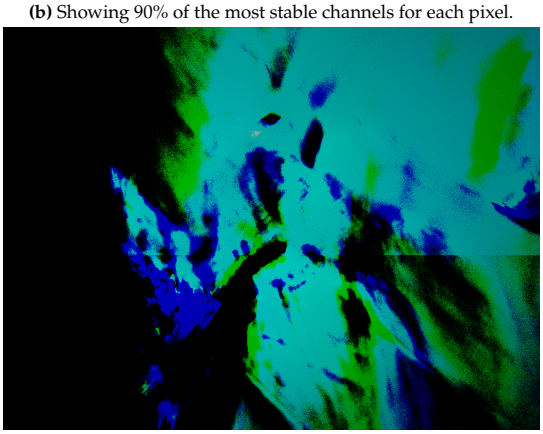
(a) Showing all channels for each pixel.



(b) Showing 90% of the most stable channels for each pixel.



(c) Showing 60% of the most stable channels for each pixel.



(d) Showing only 30% of the most stable channels for each pixel.

Figure 4.3.1: PUF picture showing only the nth stable channels for each pixel. Note that each pixel in composed of an RGB component that can be individually turned off if not considered stable. The stability is determined with the colour distance with respect to the reference picture.

4.3.2. Data Filtering Amount

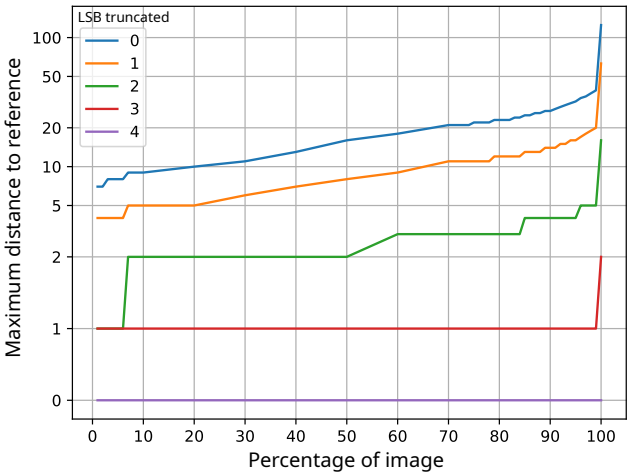
Not all the pixels/channels in the picture have the same stability, some parts of the picture are more susceptible to noise than other parts. Leaving some parts of the picture out can therefore greatly increase the stability of the PUF. However, the big downside is that there will be information missing in the final calculation. If an intrusion is only visible on the unstable pixels then it will not be detectable.

The optimal ratio of how much of the picture should be used as PUF and how much error correction is necessary needs to be determined. Figure 4.3.2a shows the maximum error of the ageing-test (Section 3.6) with different percentages of the picture taken into consideration and dropping bits of the channel. The optimal setting can be determined by crossing this table with the results from Section 3.7. Because only by knowing how much error is introduced by intrusion can an optimal configuration be determined.

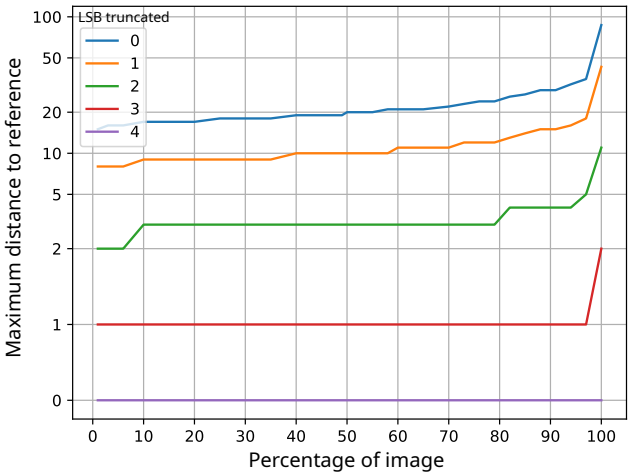
The ageing test of Section 3.6 gives some valuable insight on the maximum error that should be corrected. Using maximum colour error distance in the very last measurement as a baseline to determine how to properly handle error correction.

Taking a further look at Figure 4.3.2a, where it depicts the maximum colour distance (with various Least Significant Bit (LSB) removed and maximum amount of data considered). It can be seen that at 50% the maximum colour distance for each channel is 16 (without removing the LSBs). In order to correct this, 4 bits are necessary ($2^4 = 16$) for each byte. This is an error correction of 50%! Implementing such a strong ECC is not feasible. A workaround for this can be to remove the LSBs. When removing the two LSB (light blue line) then only a maximum distance of 2 needs to be corrected.

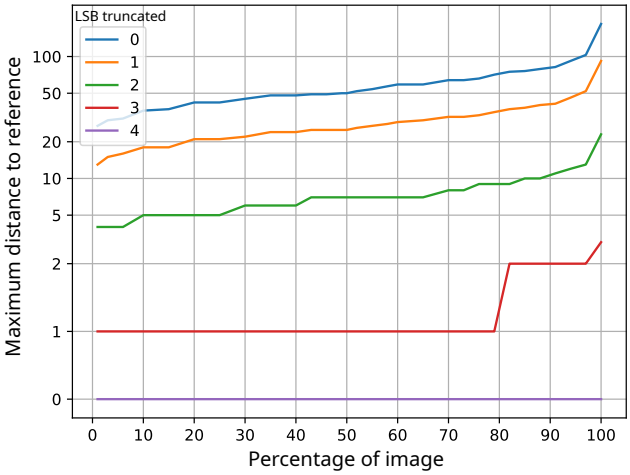
A comparison has been made to the reference picture to the picture being taken after drilling with different drills, varying the amount of stable pixels and truncating the LSBs of each individual channel (Figure 4.3.3). This figure gives the maximum colour difference between the reference and the picture taken after the drilling (considering all the different picture settings). Ideally, the difference is as big as possible. Shifting bits (different coloured lines) lessen the distance, but gives a better error correctable data (as can be seen in Figure 4.3.2a). Finding a balance between these two figures is therefore necessary. Looking at the data it can be seen that taking between 40% and 50% of the most stable channels pixels have the best result overall. This is possibly because having less pixels means having less surface area to even see the attack. And more than approximately 50% have too much noise. Also when removing the two LSBs (green line) the colour difference is still positive (although being between one and two).



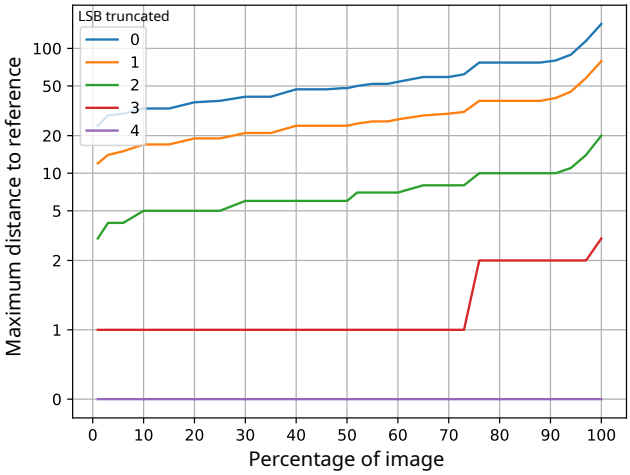
(a) Age test without enhancement and coloured.



(b) Age test without enhancement and grey-scaled.



(c) Age test with equalize adaptive and coloured.



(d) Age test with equalize adaptive and grey-scaled.

Figure 4.3.2: Maximum Colour difference of the age-test, where the X axis shows the amount (in percentage) that is being used in the comparison and the Y axis the maximum colour difference. Different lines show the LSB that are truncated.

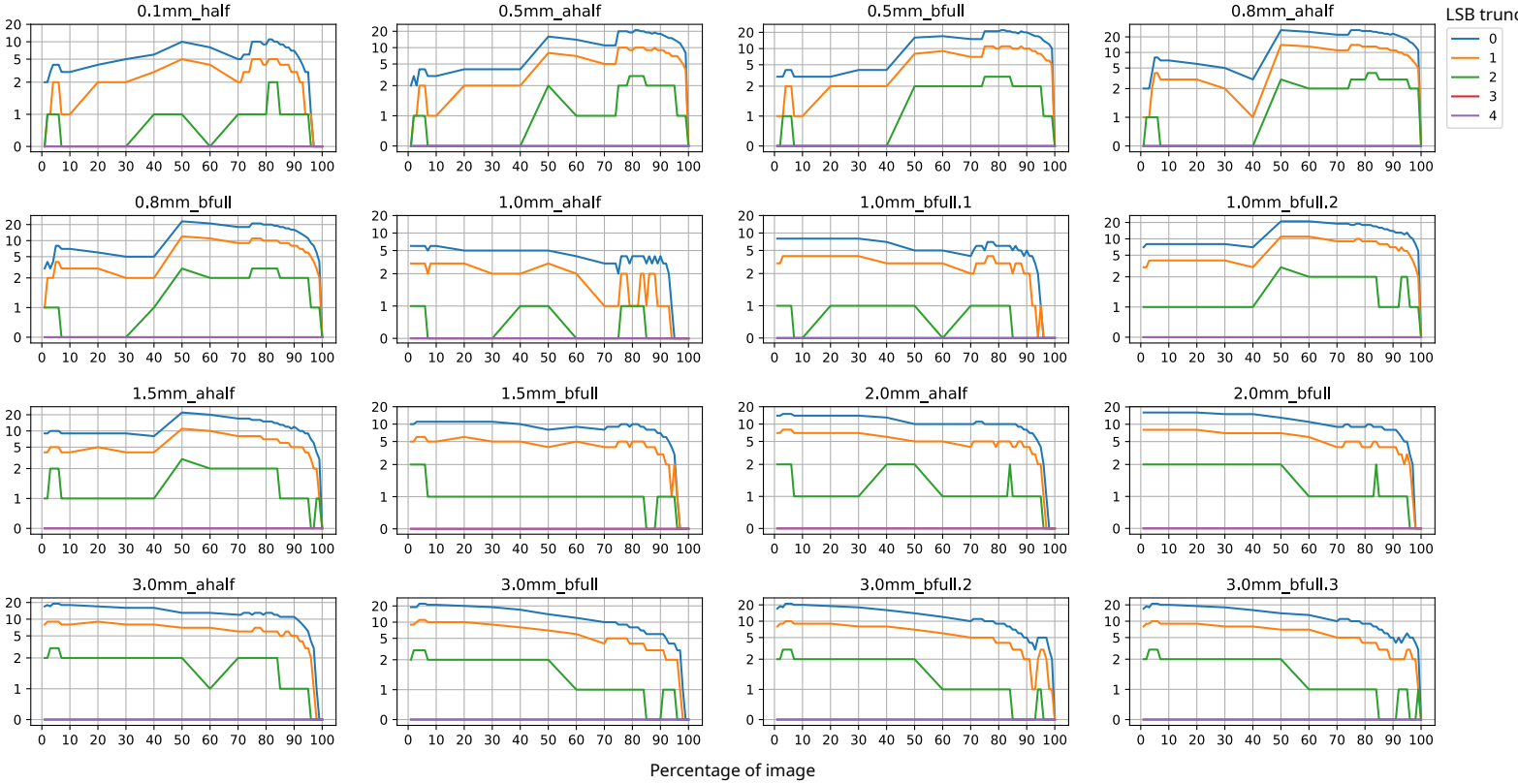


Figure 4.3.3: Maximum colour difference (Y axis) compared the reference image to the drill picture using default coloured images. The X axis show the percentage of the picture considered for the comparison.

4.3.3. Binning

Frequently fuzzy commitment is implemented with a bitwise error correction. This could (with the appropriate algorithm and configuration) correct 1011_2 (13_{10}) to 0011_2 (5_{10}). Fixing the hamming distance of 1. However, in the scheme of picture correction this is not desired. As the distance from 13_{10} to 5_{10} is 8! As discussed earlier in Section 4.3 an error is described as the distance to the correct value, not taking into account the hamming distance. Therefore, in order to have a correct error correcting scheme, only the distance to the source in decimals should matter. In order to accomplish this, binning is used without any further error correcting scheme. The binning algorithm works as follows:

- with b: A byte (range 0-255),
- v binned value,
- o offset (between $-(\frac{s}{2})$ and $\frac{s}{2}$)
- s size of the bin

$$v = \frac{b}{s}$$

$$o = \begin{cases} -\frac{b \bmod s}{2}, & \text{if } \frac{b \bmod s}{2} < \frac{s}{2} \\ \frac{b \bmod s}{2}, & \text{otherwise} \end{cases}$$

The binned value v does not need to be further error corrected as any given value within $\frac{s}{2}$ will be binned to the same value. That is however if the offset (o) has been set correctly. As the centre of the bins should correspond to the actual value before binning. This offset needs to be stored for every byte at enrolment and used when authenticating. Then the value is modified like so: $v = \lceil \frac{b-o}{s} \rceil$

4.4. Enhancement

As discussed in Section 3.4 using different post-processing steps on the image could possibly give rise to smaller detail changes. This is however a two edge sword, as noise and change through temperature change are also amplified. But the difference from drilling might be more significant. In order to evaluate this: adaptive-, local-, histogram-equalization and contrast stretching (as depicted in figure 3.4.2) are used to compute alternative images for the ageing-test (Section 3.6). Equalizing all the picture from the ageing-test and plotting the difference relative to the baseline (in Figure 4.4.1) it can be seen that the local equalization is very stable, but also has an extremely high colour difference right from the start. The other enhancement functions behave very similar to the un-enhanced variant.

When enhancing the drilling test pictures and plotting the difference before and after drilling. Then all the enhancement functions fail to provide a stable response (see figures in appendix C), with the exception of adaptive equalizing (figure 4.4.2). Here the distance to the baseline is rather huge in the whole spectrum of the test, regardless of drill size, or percentage of the image taken. However, the overall error with adaptive equalizing is also big (see figure 4.3.2c). Still using this equalization method could be beneficial since the percentage of the image used can be reduced to 20%. Then when skipping the 2 LSB a colour difference of 5 needs to be corrected. Which is equivalent to 3 bits.

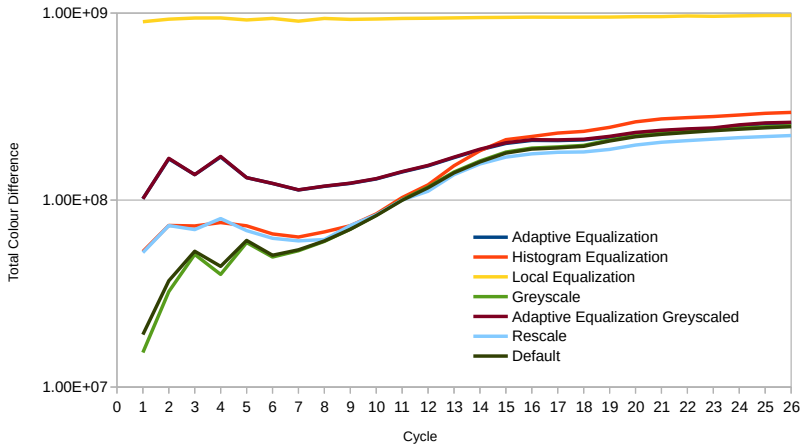


Figure 4.4.1: Total colour distance during the ageing-test cycles using: default (without any enhancement), adaptive-, histogram-, local-equalization and constrast stretching. At 22°C

4.5. Grey scaling

Using adaptive equalizing (Section 4.4) reduces the overall data needed to process. The most optimal settings for the default image are using 50% of the image and removing two LSBs. Resulting in 37.5% of the original data. Adaptive equalizing (20% of the image with the two LSBs truncated) results in 15% of the data. This value can be further reduced by converting the image to grey scale. Then only 5% of the original data is used.

It is important that when converting the image to greyscale that the weight of the colours are properly matched. Doing 1/3 of each colour channel is insufficient as it does not account for the stability of each channel. In order to properly convert the colour image to greyscale the accuracy of the channels are measured by weighting the total colour distance for each channel of the entire ageing-test (Section 3.6). This results in the final pixel value of: $B * 0.37 + G * 0.36 + R * 0.27$.

The change in diffiation is depicted in Figure 4.3.2, here it can be seen that the maximum difference becomes bigger when grey-scaling without any enhancement (Figure 4.3.2d), but stays approximately the same when using adaptive equalizing (Figure 4.3.2d). This is desired as in increase needs to be corrected. A final image looks like Figure 4.5.1.

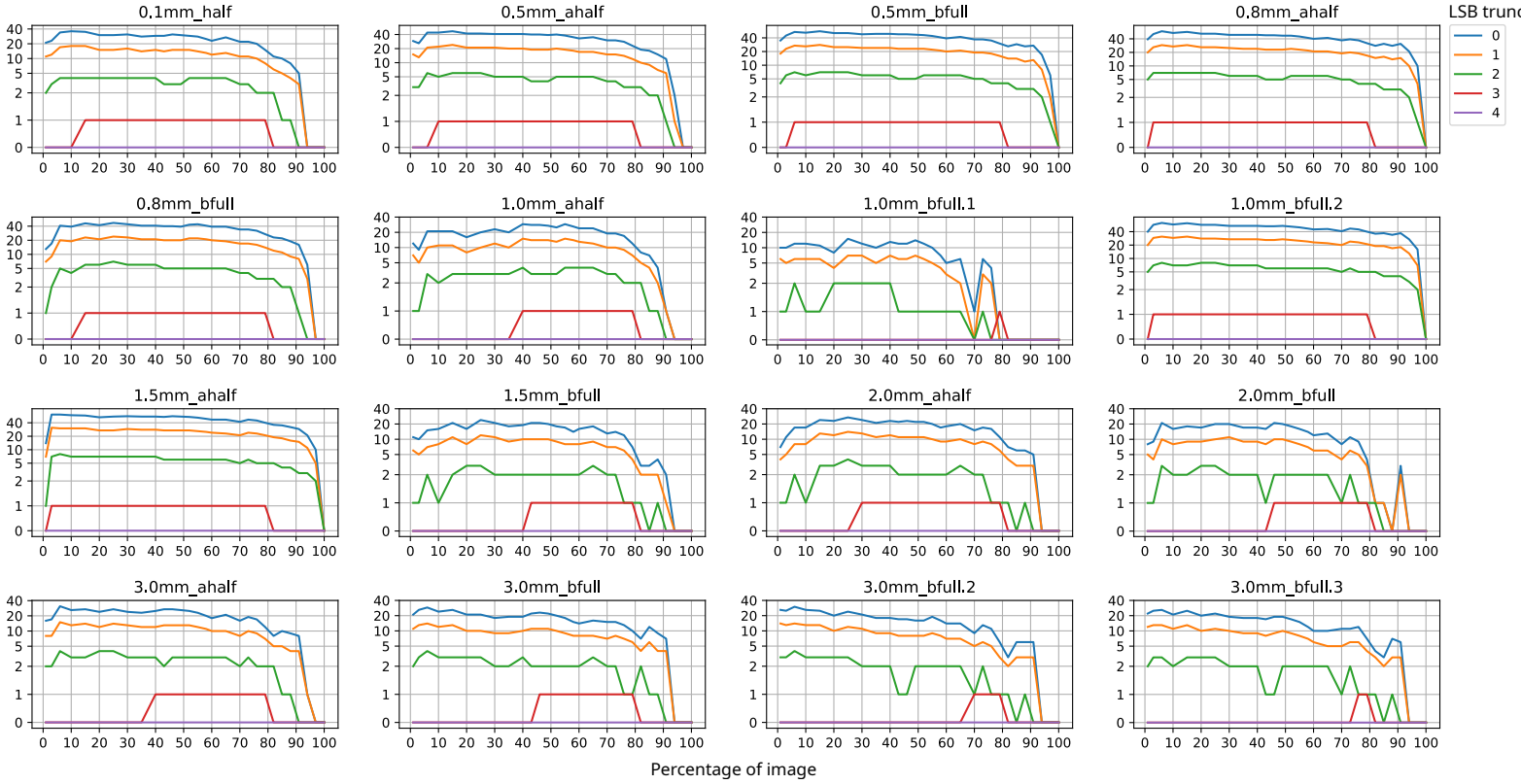


Figure 4.4.2: Colour difference when drilling using adaptive equalizing

(a) Image taken after drilling with a 1mm drill using 20% of the golden pixels with Adaptive Equalization and greyscaling applied.

(b) Image displaying invalid Sections (white) of Figure 4.5.1a that could not be error corrected using binning (amounts for over 80%).

Figure 4.5.1: Image (with invalid Sections) taken after 1mm drilling with a single light and 45s of shutter-time.

4.6. Complete Flow

As described in Section 3.6 the system is most stable at at enrollment temperature (22°C). Therefore, the enrolment and all further authentication of the system should stay at said temperature. The temperature in between may fluctuate, but the system will be more stable if the enrolment temperature holds for the entire lifetime. Then at enrolment approximately 20 pictures are taken (see Figure 3.4.1 as taking

more has diminishing returns). Then the average of those 20 samples is calculated, this picture is then equalized using the adaptive equalization. (See Section 4.4). The picture is then grey-scaled using the weights in Section 4.5. Then the most stable 20% of the picture is binned (as described in Section 4.3.3). The offset for each byte is stored. Then the image is segmented in blocks of 24 bits. For each segment a random bit stream of 24 bits is calculated and *xor'ed* with the data of the segment. This is the commitment value of this part of the image. These commitments are also stored on disk for validation. The validation process differs in that some values are loaded from disk like:

- Mask, describing which pixels to take for the computation.
- Binning offset.
- Commitment values.

Then also for each 24 bits the commitment is *xor'ed* with the corresponding segment of the picture. Then, if the values are within the size of the binning step, the original random key is recovered. The flow of the fuzzy commitment implementation can be seen in figure 4.6.1.

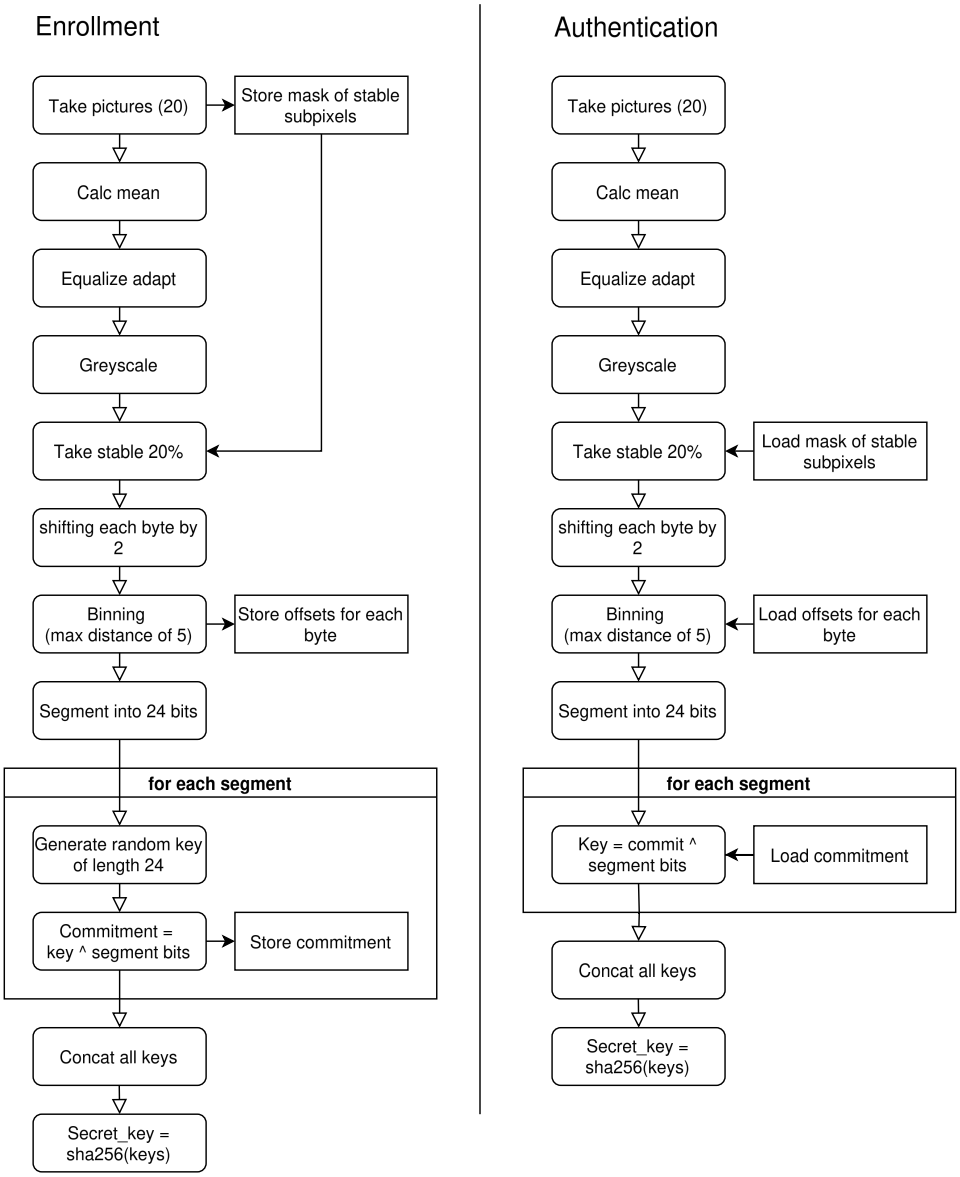


Figure 4.6.1: Flow graph of the final implementation.

4.7. Further improvements

The goal of the implementation is to benchmark the algorithm and give information on its performance with as much insights as possible.

The proof-of-concept fuzzy commitment implementation can be further improved by removing redundant instructions and store operations. This improves security as less helper data is stored on the device which could theoretically be analysed by an evil operator to extract the secret key. An implementation more closely representing fuzzy extraction can be achieved by concatting the binned values and calculating the SHA-256 sum directly from that. By doing so no commitment needs to be stored, only the offset and the mask. This is great for production but not so much for benchmarking the performance of the system since it will produce a boolean output and will not display how much of the system is faulty.

4.8. Results

A software implementation has been made of the described flow in Section 4.6 using various configurations:

- Default image without grey-scaling, 50% of data, and a bin size of 5.
- Default image without grey-scaling, 50% of data, and a bin size of 11.
- Adaptive equalized image with grey-scaling, 20% of data, and a bin size of 11.
- Adaptive equalized image with grey-scaling, 20% of data, and a bin size of 13.

The performance of the configurations can be seen in figure 4.8.1. Most of the default and adaptive configurations are able to distinguish between all drills and temperature tests. But a clear distinction can be seen between the different settings. Where the default configuration with a bin size of 5 is not reliable, but using this configuration it can be seen that the amount of correct segments decreases when the drilling size increases. Whereas other configurations are more stable throughout the different drilling sizes (see Figure 4.8.1). The equalize adaptive enhancement with a bin size of 11 also (ever so slightly) suffers from this. When going through the temperature cycles it fails to be 100% reproducible when not at the same temperature as enrollment. However, when a bin size of 13 is used the performance is somewhat the same as the default configuration with a bin size of 11. But with only 13.333% of the data stored in the device compared to the default setting (see appendix E). Because of the low data usage the computations, memory usage and storage of the adaptive equalization are much better. Furthermore, the performance is even better! Resulting in an average of approximately 17% correct segments when drilling.

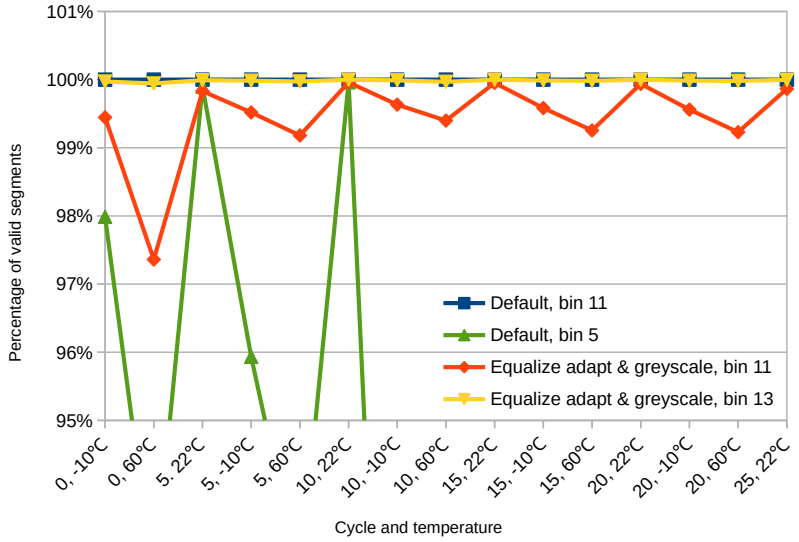
Figure 4.8.1 show that even a small hole triggers an invalid response. Note however, that this graph can be misleading. As in order to drill a 0.1mm hole predrilling has to be done. So before a 3mm predrilling has been done. Comparing this predrilling to the reference picture show a similar result in segments succeeding as with a 0.1mm drill. Not predrilling and directly using a 1mm drill could be possible, but was not tested. Looking at the default picture with a bin size of 5 (Figure 4.8.1b) starts to decrease at around 1mm. This means that even without predrilling, a noticable difference is picked-up by the algorithm at 1mm. Figure 4.8.1c also confirms this. Here only single shot images are used in comparison. And when taken the pre-drilling as reference the amount of correct segments also drops at 1mm.

Evaluating Glitter PUF against the threat Model

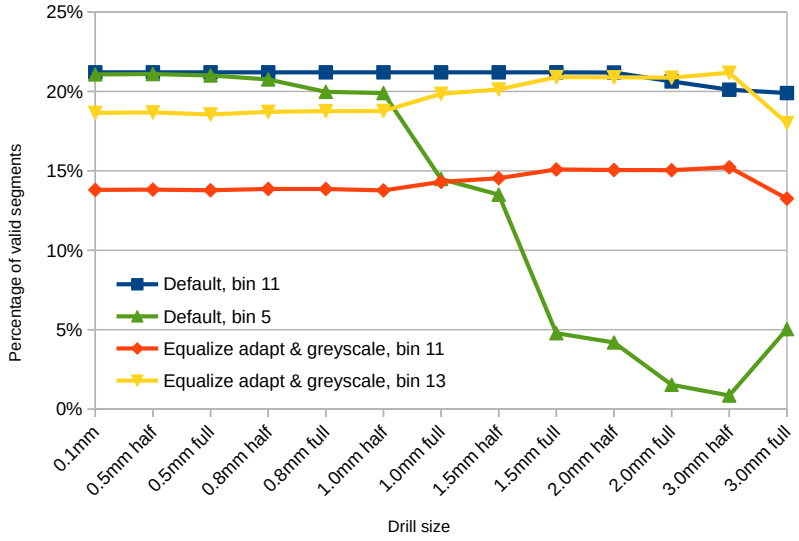
It is important to compare the results to the threat criteria divined in Section 2.2. By doing so it can be evaluated if the result is within margins and if the threats can be mitigated.

Goals

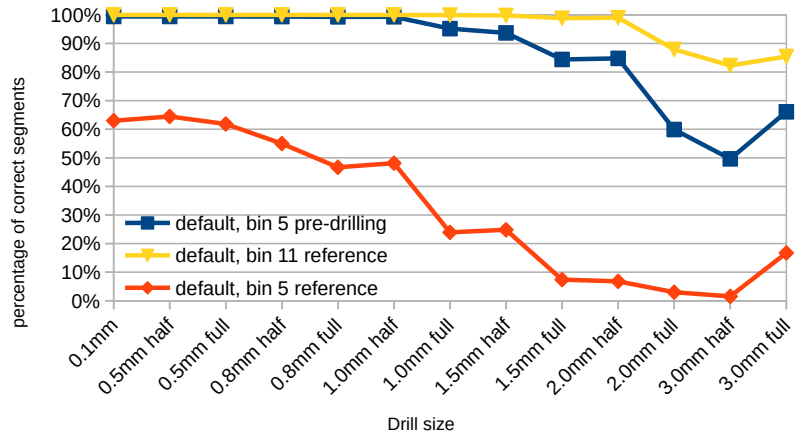
The attackers wished to modify the device or extract data from it. In order to do so the adversary must first access the electronics of the PCB. Being that the entire device is enclosed with glitter PUF one must first penetrate it. After penetration a small wire must make contact in-order to do some kind of modification or data extraction from the device. A wire can be easily made with a diameter of 0.5mm. Drilling a hole of said diameter triggers an incorrect PUF response and could therefore be said that the protection is sufficient.



(a) Segments percentage succeeding when going though temperature cycles



(b) Segments percentage succeeding when drilling with various sizes



(c) Segments percentage succeeding when drilling with various sizes using only a single picture for comparance. And different reference point

Figure 4.8.1: Results of the prototype code described in Section 4.6 using the average of the glitter PUF settings.

Capabilities
Physical access to the device is granted, and as stated before the anti-tamper technique is well-equipped to deal with it. However, if the adversary is equipped with a x-ray device. Then this could (possibly) show the layout of connections. Furthermore, if the glitter particles are big enough then they could also be indexed by the x-ray. Having plenty of smaller glitter particles is therefore recommended.

5

Conclusion & Future Work

This section describes the conlusion of this paper, and possible future research that is required to further asses glitter PUF. Also concluding the findings of the experimentation and implementation.

5.1. Conclusion

The nature of Glitter PUF is quite unique and requires an exact approach to categorize and test its security meaningfulness. By characterizing its properties at different temperatures and ageing-tests. Ageing test are performed over 23 temperature cycles to see the resulting error. These tests show the behavior of the device while in operation and is used to compare the to the results while being tampered with. Analyzing these operation characteristics it can be said that the PUF inherently (like many other PUFs) are unstable in nature. Temperature deviations have a big effect on the image produced, that could result in a false positive.

Filtering the (sub)pixels and baking the PUF to stabilize it, is necessary before enrollment. The filtering is best done by going through more then 20 temperature cycles (moving from 60°C to -10°C) and taking the most stable 20% of the image.

Tampering with the device by means of drilling was achieved by using various drill sizes. During these experiments, it became apparent that error produced by the drill is local to a region of the image.

Afterward it's best to use adaptive equalization enhancement; grey-scaling the image (using weighted values for the RGB component based on the stability); removing the 2 LSB for each subpixel. This compresses the data usage to 5% of the original data. While keeping all the details to properly detect drillings from 0.1mm to 3mm. Then afterward Fuzzy Commitment can be used on segments of the image. It's important to segment the image, as the error introduced by drilling is local to a region. Segmentation of 24 bits is used, each segment contributing to the secret key. The results show that the system can easily detect each drill when the picture from before the temperature test is used. However, this could also be because of the pre-drilling done beforehand. This pre-drilling is however necessary to drill the smaller drills. As the drill is otherwise not long enough to penetrate through the enclosure and reach the resin. Regardless, when comparing the drilled holes with as reference the pre-drilling a drill of 1mm hole is also detectable. Supplementary tests are however necessary in-order to further validate the drilling capabilities. As many tests are unfortunately not done because of timing constrains (as discussed in Section 5.2).

5.2. Future Work

The research done in the article was primary focused on inventorization of the glitter PUF; characterization techniques for its improvements. In order to fully validate the system for practical use a number of steps need to be taken:

- **Drilling at different angles**: in this research only drilling straight down has been tested. Different angles may show different behaviour.

- **Drilling at different places (outside the direct FoV of the camera)**: In Section 3.7 only drilling directly above the camera has been tested. Drilling outside the FoV of the camera must be tested in order to evaluate the reflective properties of the glitters.
- **Researching more advanced Fuzzy schemes**: The current prototype uses fuzzy commitment. This is however, not the most cunning scheme. As more advanced techniques exist. Utilizing one of those could properly enhance the speed, security and performance of the system.
- **Optimal camera and LED placement**: As can be seen in figure 3.2.1a the camera and LEDs are placed far away from each other. Another configuration as discussed in Section 3.8 could be considered.
- **Optimal enclosure shape**: The shape of the enclosure was not considered in this research. This is however vital for a proper secure implementation.
- **Unpredictability of image data**: Ideally the image data gathered from the sensor is completely random (but stable) for an individual PUF. This is however possibly not the case, and the image data might be predictable and thus subject to a brute-force attack. The image pattern is not analyzed, but should glitter PUF be used in a security critical application. Then it should.
- **Testing Diamond PUF**: Unfortunately though time-constrains the Diamond PUF in Section 3.8 has not been tested. Comparing this PUF to the Glitter PUF might show interesting results.
- **Repudiation tests**: An adversary might try to clone, or reproduce the image by (for example) carefully removing the PUR of the device, making modifications and then reapplying the same PUR back in place. These kinds of test must be done in order to evaluate the total security of the Glitter PUF.
- **Intra distance testing**: All the experimentation in this research have been done on a single PUF. This is great for characterization of the device. But when deploying multiple systems the distance between the responses must also be significant. So that one PUFs response is not near another one of the same family.
- **Frequency domain**: The error correction and filtering of the data used in the picture have been done in the colour and position domain of the image. Another approach by looking at the frequencies in the image could be interesting.
- **Optimal resin and particle material**: Alternatives for the resin used in this research, even as the reflective particles are not researched. A more optimal solution might exist.

References

[1] Vincent Immler et al. "B-TREPID: Batteryless tamper-resistant envelope with a PUF and integrity detection". In: *2018 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. 2018, pp. 49–56. DOI: 10.1109/HST.2018.8383890.

[2] Vincent Immler et al. "Secure Physical Enclosures from Covers with Tamper-Resistance". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2019.1 (2018), 51–96. DOI: 10.13154/tches.v2019.i1.51-96. URL: https://tches.iacr.org/index.php/TCHES/article/view/7334.

[3] Matin Barekatain, Hai Liu, and Eun Sok Kim. "Wireless and Battery-Less Tamper Detection With Pyroelectric Energy Converter and High-Overtone Bulk Acoustic Resonator". In: *IEEE Sensors Journal* 22.14 (2022), pp. 14639–14646. DOI: 10.1109/JSEN.2022.3182940.

[4] Enisa. *ENISA Threat Landscape 2023*. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023.

[5] Ömer Aslan et al. "A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions". In: *Electronics* 12 (Mar. 2023), pp. 1–42. DOI: 10.3390/electronics12061333.

[6] Md Mallick and Rishab Nath. "Navigating the Cyber security Landscape: A Comprehensive Review of Cyber-Attacks, Emerging Trends, and Recent Developments". In: (Feb. 2024).

[7] G4S. *World Security Report*. URL: https://www.worldsecurityreport.com/key-findings.

[8] Enisa. *Hardware Threat Landscape and Good Practice Guide*. URL: https://www.enisa.europa.eu/publications/hardware-threat-landscape.

[9] Farinaz Koushanfar. "Provably Secure Active IC Metering Techniques for Piracy Avoidance and Digital Rights Management". In: *IEEE Transactions on Information Forensics and Security* 7.1 (2012), pp. 51–63. DOI: 10.1109/TIFS.2011.2163307.

[10] Rajat Subhra Chakraborty and Swarup Bhunia. "Hardware protection and authentication through netlist level obfuscation". In: *2008 IEEE/ACM International Conference on Computer-Aided Design*. 2008, pp. 674–677. DOI: 10.1109/ICCAD.2008.4681649.

[11] Clément Gaine et al. "Active Shielding Against Physical Attacks by Observation and Fault Injection: ChaXa". In: *Journal of Hardware and Systems Security* 7 (Feb. 2023), pp. 1–10. DOI: 10.1007/s41635-023-00131-5.

[12] Daniel-Ciprian Vasile and Paul Svasta. "Protecting the Secrets: Advanced Technique for Active Tamper Detection Systems". In: (2019), pp. 212–215. DOI: 10.1109/SIITME47687.2019.8990877.

[13] Halit Eren and Lucas D Sandor. "Fringe-Effect Capacitive Proximity Sensors for Tamper Proof Enclosures". In: *2005 Sensors for Industry Conference*. 2005, pp. 22–26. DOI: 10.1109/SICON.2005.257863.

[14] Tahoura Mosavirik et al. "Silicon Echoes: Non-Invasive Trojan and Tamper Detection using Frequency-Selective Impedance Analysis". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2023.4 (2023), 238–261. DOI: 10.46586/tches.v2023.i4.238-261. URL: https://tches.iacr.org/index.php/TCHES/article/view/11165.

[15] Fengchao Zhang, Andrew Hennessy, and Swarup Bhunia. "Robust counterfeit PCB detection exploiting intrinsic trace impedance variations". In: *2015 IEEE 33rd VLSI Test Symposium (VTS)*. 2015, pp. 1–6. DOI: 10.1109/VTS.2015.7116294.

[16] J. Alex Halderman et al. "Lest we remember: cold-boot attacks on encryption keys". In: *Commun. ACM* 52.5 (2009), 91–98. ISSN: 0001-0782. DOI: 10.1145/1506409.1506429. URL: https://doi.org/10.1145/1506409.1506429.

[17] Paul Staat et al. "Anti-Tamper Radio: System-Level Tamper Detection for Computing Systems". In: *2022 IEEE Symposium on Security and Privacy (SP)*. 2022, pp. 1722–1736. DOI: 10.1109/SP46214.2022.9833631.

[18] Jorge Guajardo. "Physical Unclonable Functions (PUFs)". In: *Encyclopedia of Cryptography and Security*. Ed. by Henk C. A. van Tilborg and Sushil Jajodia. Boston, MA: Springer US, 2011,

pp. 929–934. ISBN: 978-1-4419-5906-5. DOI: 10.1007/978-1-4419-5906-5_912. URL: https://doi.org/10.1007/978-1-4419-5906-5_912.

[19] Roel Maes. *Physically Unclonable Functions: Constructions, Properties and Applications*. Springer Publishing Company, Incorporated, 2013. ISBN: 3642413943.

[20] Jorge Guajardo et al. "FPGA Intrinsic PUFs and Their Use for IP Protection". In: *Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems*. CHES '07. Vienna, Austria: Springer-Verlag, 2007, 63–80. ISBN: 9783540747345. DOI: 10.1007/978-3-540-74735-2_5. URL: https://doi.org/10.1007/978-3-540-74735-2_5.

[21] Ulrich Rührmair, Heike Busch, and Stefan Katzenbeisser. "Strong PUFs: Models, Constructions, and Security Proofs". In: Oct. 2010, pp. 79–96. ISBN: 978-3-642-14451-6. DOI: 10.1007/978-3-642-14452-3_4.

[22] Huansheng Ning et al. "Physical unclonable function: architectures, applications and challenges for dependable security". In: *IET Circuits, Devices & Systems* 14.4 (2020), pp. 407–424. DOI: https://doi.org/10.1049/iet-cds.2019.0175. eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-cds.2019.0175. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-cds.2019.0175.

[23] Charis Mesaritakis et al. "Physical Unclonable Function based on a Multi-Mode Optical Waveguide". In: *Scientific Reports* 8 (June 2018). DOI: 10.1038/s41598-018-28008-6.

[24] Min Kim et al. "Revisiting silk: a lens-free optical physical unclonable function". In: *Nature Communications* 13 (Jan. 2022). DOI: 10.1038/s41467-021-27278-5.

[25] J.W. Lee et al. "A technique to build a secret key in integrated circuits for identification and authentication applications". In: *2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No.04CH37525)*. 2004, pp. 176–179. DOI: 10.1109/VLSIC.2004.1346548.

[26] Tarek A. Idriss, Haytham A. Idriss, and Magdy A. Bayoumi. "A Lightweight PUF-Based Authentication Protocol Using Secret Pattern Recognition for Constrained IoT Devices". In: *IEEE Access* 9 (2021), pp. 80546–80558. DOI: 10.1109/ACCESS.2021.3084903.

[27] Rajat Subhra Chakraborty and Swarup Bhunia. "HARPOON: An Obfuscation-Based SoC Design Methodology for Hardware Protection". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 28.10 (2009), pp. 1493–1502. DOI: 10.1109/TCAD.2009.2028166.

[28] Ravikanth Pappu et al. "Physical One-Way Functions". In: *Science* 297.5589 (2002), pp. 2026–2030. DOI: 10.1126/science.1074376. eprint: https://www.science.org/doi/pdf/10.1126/science.1074376. URL: https://www.science.org/doi/abs/10.1126/science.1074376.

[29] Benedikt Wigger et al. "Using unique surface patterns of injection moulded plastic components as an image based Physical Unclonable Function for secure component identification". In: *Scientific Reports* 8 (Mar. 2018). DOI: 10.1038/s41598-018-22876-8.

[30] Abayomi Jegede et al. "State of the Art in Biometric Key Binding and Key Generation Schemes". In: *International Journal of Communication Networks and Information Security* 9 (Dec. 2017), pp. 333–344. DOI: 10.17762/ijcnis.v9i3.2388.

[31] Ari Juels and Martin Wattenberg. "A Fuzzy Commitment Scheme". In: *Proceedings of the ACM Conference on Computer and Communications Security* 1 (Dec. 1999). DOI: 10.1145/319709.319714.

[32] Tanya Ignatenko and Frans M. J. Willems. "Information Leakage in Fuzzy Commitment Schemes". In: *IEEE Transactions on Information Forensics and Security* 5.2 (2010), pp. 337–348. DOI: 10.1109/TIFS.2010.2046984.

[33] Sonam Chauhan and Amit Kumar Sharma. "Improved fuzzy commitment scheme". In: *International Journal of Information Technology* (2019), pp. 1–11. URL: https://api.semanticscholar.org/CorpusID:59551964.

[34] Emile J. C. Kelkboom et al. "Preventing the Decodability Attack Based Cross-Matching in a Fuzzy Commitment Scheme". In: *IEEE Transactions on Information Forensics and Security* 6.1 (2011), pp. 107–121. DOI: 10.1109/TIFS.2010.2091637.

[35] DaeHun Nyang and KyungHee Lee. "Fuzzy Face Vault: How to Implement Fuzzy Vault with Weighted Features". In: *Universal Acess in Human Computer Interaction. Coping with Diversity*. Ed. by Constantine Stephanidis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 491–496. ISBN: 978-3-540-73279-2.

[36] A. Juels and M. Sudan. "A fuzzy vault scheme". In: (2002), pp. 408–. DOI: 10.1109/ISIT.2002.1023680.

[37] Yevgeniy Dodis et al. *Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data*. Cryptology ePrint Archive, Paper 2003/235. https://eprint.iacr.org/2003/235. 2003. URL: https://eprint.iacr.org/2003/235.

[38] Jeroen Delvaux et al. "Efficient Fuzzy Extraction of PUF-Induced Secrets: Theory and Applications". In: *Cryptographic Hardware and Embedded Systems – CHES 2016*. Ed. by Benedikt Gierlichs and Axel Y. Poschmann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 412–431. ISBN: 978-3-662-53140-2.

[39] Kaini Chen et al. "Secure Sketch and Fuzzy Extractor with Imperfect Randomness: An Information-Theoretic Study". In: *Information and Communications Security*. Ed. by Cristina Alcaraz et al. Cham: Springer International Publishing, 2022, pp. 128–147. ISBN: 978-3-031-15777-6.

[40] Christoph Bösch et al. "Efficient Helper Data Key Extractor on FPGAs". In: *Cryptographic Hardware and Embedded Systems – CHES 2008*. Ed. by Elisabeth Oswald and Pankaj Rohatgi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 181–197. ISBN: 978-3-540-85053-3.

[41] Roel Maes, Pim Tuyls, and Ingrid Verbauwhede. "A soft decision helper data algorithm for SRAM PUFs". In: *2009 IEEE International Symposium on Information Theory*. 2009, pp. 2101–2105. DOI: 10.1109/ISIT.2009.5205263.

[42] Matthias Hiller, Ludwig Kürzinger, and Georg Sigl. "Review of error correction for PUFs and evaluation on state-of-the-art FPGAs". In: *Journal of Cryptographic Engineering* 10 (Sept. 2020). DOI: 10.1007/s13389-020-00223-w.

[43] Roel Maes et al. "Secure key generation from biased PUFs: extended version". In: *Journal of Cryptographic Engineering* 6 (2016), pp. 121 –137. URL: https://api.semanticscholar.org/CorpusID:10233259.

[44] Daihyun Lim et al. "Extracting secret keys from integrated circuits". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 13.10 (2005), pp. 1200–1205. DOI: 10.1109/TVLSI.2005.859470.

[45] Yuan Cao et al. "CMOS Image Sensor Based Physical Unclonable Function for Coherent Sensor-Level Authentication". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 62.11 (2015), pp. 2629–2640. DOI: 10.1109/TCSI.2015.2476318.

[46] Yuan Cao et al. "CMOS image sensor based physical unclonable function for smart phone security applications". In: *2014 International Symposium on Integrated Circuits (ISIC)*. 2014, pp. 392–395. DOI: 10.1109/ISICIR.2014.7029496.

[47] Saloomeh Shariati et al. "Analysis and experimental evaluation of image-based PUFs". In: *Journal of Cryptographic Engineering* 2 (Oct. 2012). DOI: 10.1007/s13389-012-0041-3.

[48] Electrolube. *UR5645 polyurethane Resin Datasheet*. URL: https://electrolube.com/app/uploads/2020/03/UR5645-1.pdf.

[49] C. E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[50] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861.

A

Taking Picture on Raspberry Pi

Below is the photo command used on the RPi for taken a picture. Note that the settings are chosen to be as neutral and as close to the real world as possible.

```
raspistill -w 3280 -h 2464 -o \"{}\".jpg -v -t 1 -set -n -sh 0 -co 0 -br 50 -sa 0 -ISO 0 -ex off -fli off -awb off -awbg 2.23,1 -ifx none -ss {} -drc off -st -ag 1 -dg 1 -e jpg -q 100 --raw
```

Listing A.0.1: Command used for taking photos

B

Optimal Glitter PUF Picture Settings

The table below is used to determine the best picture settings for the glitter PUF. The data inside the table is the normalized score, using shannon entropy and SSIM. The higher the score, the more stable and the more information a configuration has. The LED number indicate with LEDs are on for that configuration. LED 0 being the LED closest to the camera, and 4 the furthers (see figure 3.2.1a).

	Shutter Speed (s)											
LED	.5	1	2	4	6	8	10	12	14	16	18	20
0	36.40%	54.35%	66.47%	76.97%	82.19%	85.36%	87.58%	89.21%	89.16%	89.80%	90.79%	91.31%
1	31.22%	41.64%	58.41%	69.62%	75.48%	79.39%	82.33%	84.50%	86.21%	87.59%	88.75%	89.74%
2	14.84%	32.08%	41.47%	58.59%	65.00%	69.42%	72.55%	75.25%	77.32%	79.08%	80.73%	82.09%
3	0.85%	13.08%	31.93%	40.05%	51.58%	57.23%	61.17%	63.95%	66.08%	68.27%	70.02%	71.46%
4	0.00%	2.29%	17.90%	32.52%	35.02%	43.87%	50.01%	54.09%	56.59%	58.69%	60.60%	62.20%
10	46.38%	61.06%	72.94%	82.84%	87.17%	89.71%	89.71%	90.70%	91.31%	91.95%	92.23%	92.46%
20	42.14%	57.99%	69.99%	80.37%	85.08%	87.94%	89.37%	89.51%	90.38%	91.07%	91.48%	91.86%
21	32.68%	51.54%	64.28%	75.39%	80.86%	84.40%	86.87%	88.70%	90.12%	91.25%	92.17%	92.90%
30	39.38%	56.21%	68.17%	78.72%	83.63%	86.65%	88.73%	88.73%	89.53%	90.40%	91.03%	91.47%
31	31.56%	47.24%	61.56%	72.65%	78.28%	82.11%	84.75%	86.75%	88.29%	89.53%	90.58%	91.44%
32	25.90%	34.02%	51.88%	64.62%	70.68%	74.86%	77.85%	80.35%	82.34%	83.92%	85.27%	86.44%
40	38.18%	55.36%	67.43%	77.92%	82.96%	86.04%	88.17%	89.08%	89.12%	90.01%	90.66%	91.21%
41	31.40%	45.02%	60.19%	71.26%	77.02%	80.90%	83.65%	85.71%	87.32%	88.60%	89.69%	90.61%
42	21.94%	32.96%	48.10%	62.18%	68.44%	72.50%	75.72%	78.11%	80.24%	81.93%	83.35%	84.58%
43	7.07%	26.92%	33.87%	52.44%	59.73%	64.13%	67.52%	70.14%	72.35%	74.25%	75.87%	77.26%
210	49.45%	63.78%	75.31%	84.72%	88.79%	89.70%	90.76%	91.69%	92.14%	92.52%	92.77%	92.93%
310	47.70%	62.39%	74.09%	83.76%	87.96%	89.31%	90.20%	91.03%	91.61%	92.12%	92.44%	92.69%
320	44.06%	59.52%	71.43%	81.57%	86.08%	88.83%	89.04%	90.21%	90.90%	91.30%	91.78%	92.17%
321	35.24%	54.10%	66.56%	77.28%	82.66%	85.97%	88.28%	89.99%	90.93%	91.52%	92.69%	93.32%
410	47.22%	61.71%	73.57%	83.33%	87.58%	89.65%	89.95%	90.81%	91.48%	92.01%	92.26%	92.56%
420	43.29%	58.84%	70.70%	81.01%	85.60%	88.39%	88.87%	89.80%	90.45%	91.19%	91.58%	91.95%
421	33.85%	53.06%	65.54%	76.42%	81.83%	85.23%	87.59%	89.35%	90.70%	91.43%	92.44%	93.10%
430	40.78%	57.04%	69.04%	79.47%	84.27%	87.19%	89.21%	89.06%	89.75%	90.54%	90.97%	91.53%
431	31.75%	49.62%	62.87%	73.96%	79.50%	83.17%	85.69%	87.60%	89.07%	90.25%	91.23%	92.04%
432	29.62%	36.72%	55.20%	66.93%	72.87%	76.89%	79.91%	82.25%	84.07%	85.55%	86.81%	87.87%
3210	50.62%	65.04%	76.18%	85.46%	89.39%	89.87%	91.09%	91.85%	92.42%	92.73%	92.95%	93.12%
4210	50.26%	64.52%	75.77%	85.11%	89.09%	89.70%	90.84%	91.65%	92.24%	92.63%	92.80%	93.03%
4310	48.60%	63.08%	74.63%	84.18%	88.30%	89.14%	90.38%	91.25%	91.79%	92.30%	92.57%	92.73%
4320	45.19%	60.14%	72.14%	82.11%	86.52%	89.20%	89.24%	90.24%	91.04%	91.56%	91.92%	92.24%
4321	36.96%	55.37%	67.55%	78.14%	83.41%	86.62%	88.84%	90.50%	90.91%	91.62%	92.41%	93.04%
43210	51.07%	65.45%	76.65%	85.78%	89.64%	90.11%	91.20%	91.93%	92.53%	92.76%	93.01%	93.00%
-	3.89%	3.91%	3.90%	3.78%	3.61%	3.51%	3.38%	3.25%	2.83%	2.29%	2.18%	2.38%

(a) First part of the table

	Shutter Speed (s)								
LED	22	24	26	28	30	35	40	45	50
0	91.94%	92.39%	92.58%	93.01%	93.16%	93.77%	93.98%	94.08%	94.24%
1	90.61%	91.33%	91.96%	92.51%	93.01%	93.97%	94.74%	95.32%	95.79%
2	83.33%	84.36%	85.25%	86.09%	86.86%	88.41%	89.68%	90.72%	91.59%
3	72.96%	74.23%	75.35%	76.36%	77.26%	79.36%	81.14%	82.58%	83.80%
4	63.51%	64.82%	66.19%	67.35%	68.32%	70.34%	72.33%	73.91%	75.26%
10	92.87%	93.07%	93.15%	93.24%	93.43%	93.34%	93.11%	92.74%	91.96%
20	92.38%	92.80%	92.92%	93.18%	93.32%	93.71%	93.67%	93.64%	93.42%
21	93.63%	94.18%	94.63%	95.04%	95.38%	95.56%	96.07%	96.04%	96.26%
30	92.17%	92.31%	92.63%	92.96%	93.12%	93.71%	93.84%	93.86%	93.77%
31	92.21%	92.84%	93.37%	93.84%	94.26%	95.08%	95.69%	96.12%	96.44%
32	87.50%	88.36%	89.11%	89.80%	90.42%	91.70%	92.70%	93.51%	94.17%
40	91.86%	92.15%	92.68%	92.87%	93.13%	93.58%	93.90%	93.83%	93.88%
41	91.42%	92.08%	92.65%	93.15%	93.60%	94.47%	95.15%	95.65%	96.02%
42	85.70%	86.61%	87.41%	88.15%	88.82%	90.23%	91.31%	92.21%	92.95%
43	78.68%	79.84%	80.86%	81.77%	82.59%	84.36%	85.80%	86.98%	87.99%
210	93.02%	93.13%	93.04%	93.06%	93.05%	92.78%	92.05%	90.85%	89.13%
310	92.91%	93.04%	93.15%	93.18%	93.17%	92.88%	92.60%	91.84%	90.76%
320	92.61%	92.91%	93.11%	93.31%	93.41%	93.65%	93.54%	93.29%	92.99%
321	94.10%	94.68%	94.79%	95.07%	95.21%	95.27%	95.59%	95.65%	95.69%
410	92.82%	92.95%	93.18%	93.13%	93.16%	93.01%	92.80%	92.26%	91.25%
420	92.34%	92.58%	92.83%	93.14%	93.26%	93.55%	93.52%	93.34%	93.08%
421	93.96%	94.38%	94.63%	95.13%	95.24%	95.07%	95.44%	95.48%	95.62%
430	92.00%	92.24%	92.60%	92.77%	93.20%	93.50%	93.57%	93.58%	93.41%
431	92.77%	93.34%	93.83%	94.27%	94.65%	95.37%	95.88%	96.23%	96.21%
432	88.86%	89.66%	90.34%	90.96%	91.52%	92.66%	93.55%	94.25%	94.82%
3210	93.04%	93.10%	92.98%	92.94%	92.92%	92.22%	91.34%	89.79%	87.62%
4210	93.06%	93.10%	93.00%	92.97%	92.83%	92.37%	91.61%	90.10%	88.39%
4310	92.86%	92.91%	93.01%	92.97%	93.05%	92.66%	92.34%	91.34%	89.76%
4320	92.52%	92.87%	93.12%	93.20%	93.30%	93.14%	93.27%	92.99%	92.41%
4321	93.63%	94.21%	94.56%	94.85%	94.94%	95.08%	95.33%	95.43%	95.33%
43210	93.16%	93.11%	92.92%	92.85%	92.70%	91.91%	90.97%	88.98%	86.83%
-	2.00%	1.86%	1.77%	1.64%	1.74%	1.58%	1.18%	1.86%	1.89%

(b) 2nd part of the table

Table B.0.1: Shannon Entropy weighted with the SSIM then normalized of the glitter PUF prototype.

C

Picture Enhancement After Drilling

Colour difference when drilling using enhancement features on the images. Note that the figures below show that using any of these enhancement techniques result in a unstable image. Where it is not possible to detect a drill reliably.

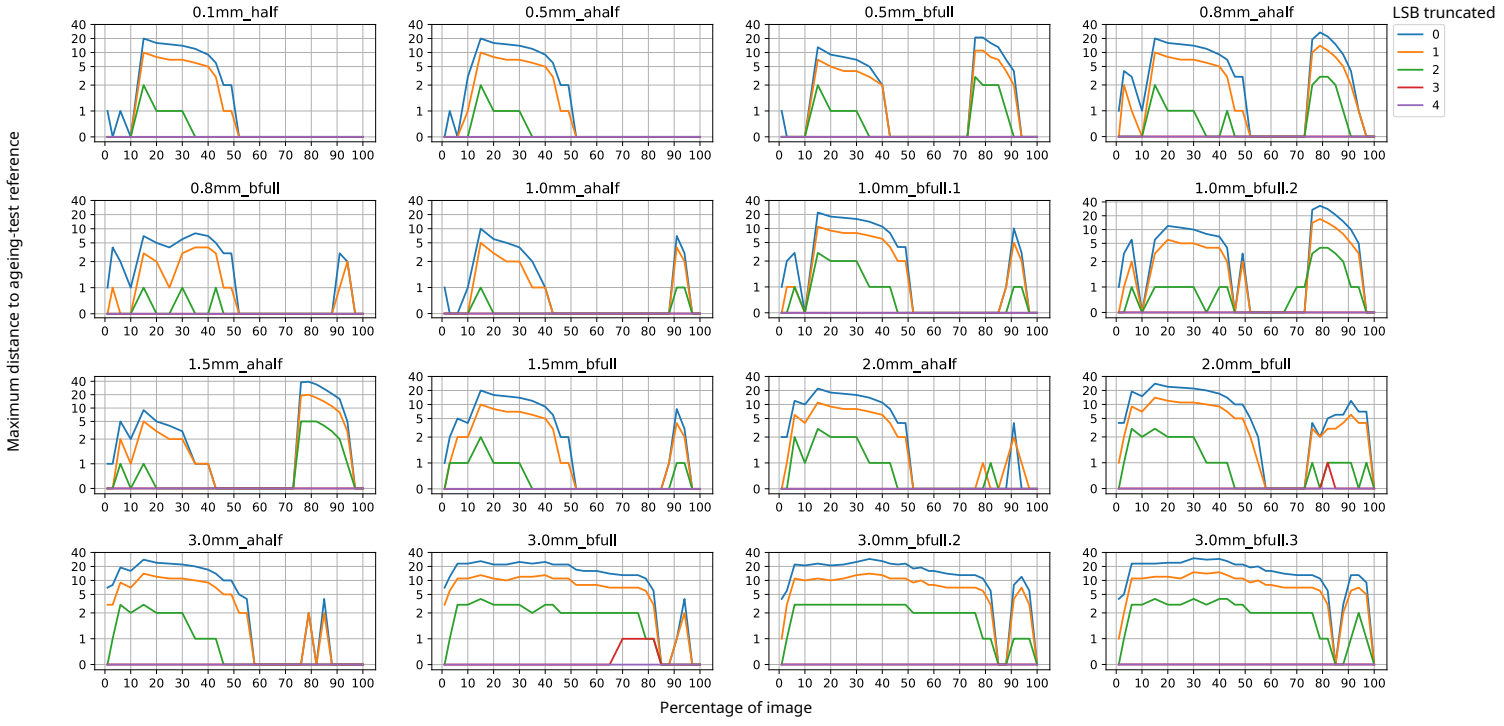


Figure C.0.1: Colour difference when drilling using histogram equalization

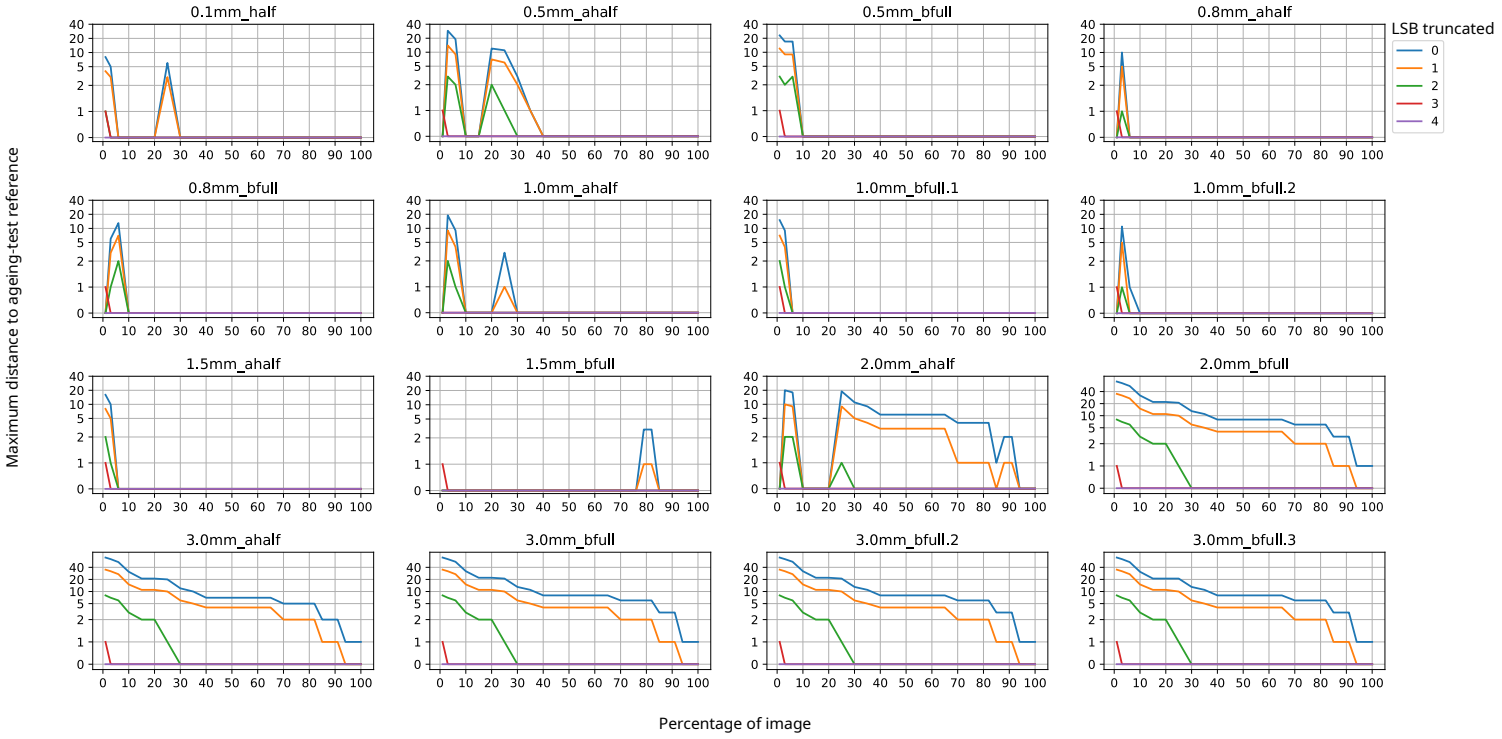


Figure C.0.2: Colour difference when drilling using local equalization

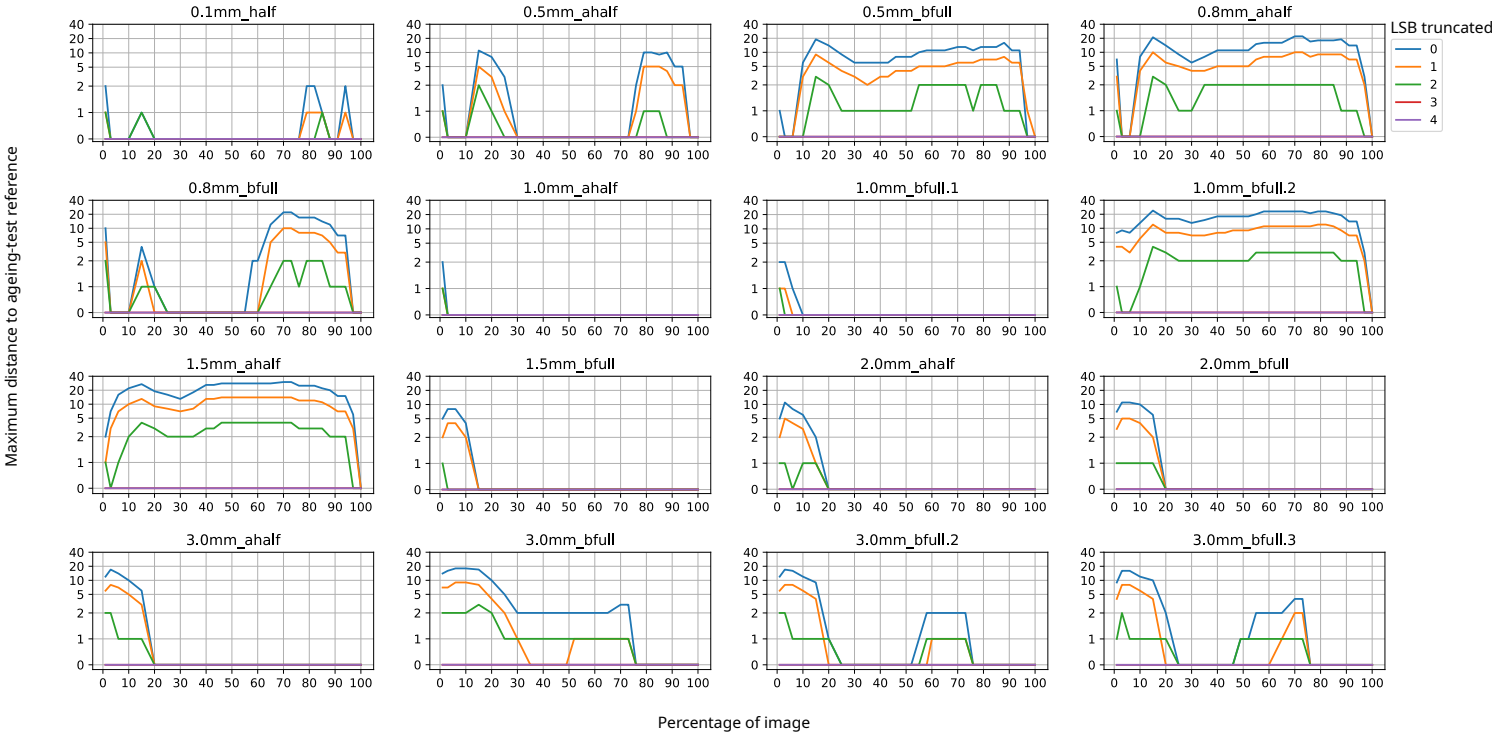
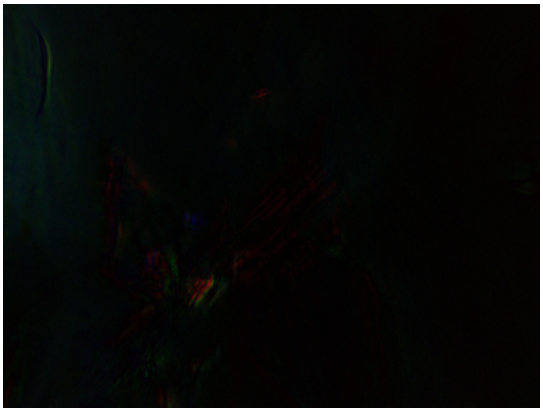


Figure C.0.3: Colour difference when drilling using constrast stretching

D

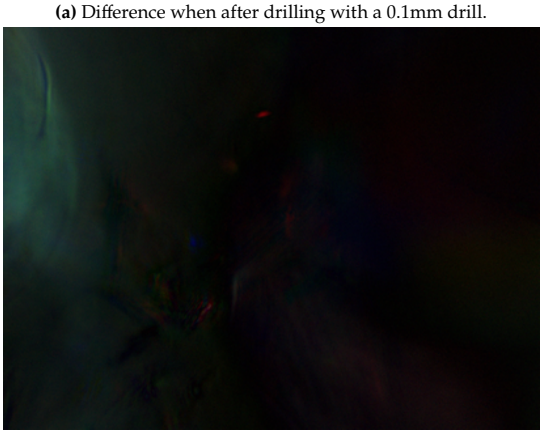
Impact of Drilling on Image

Pictures below show the difference (amplified 25 times) when drilling with diffrent drill sizes.

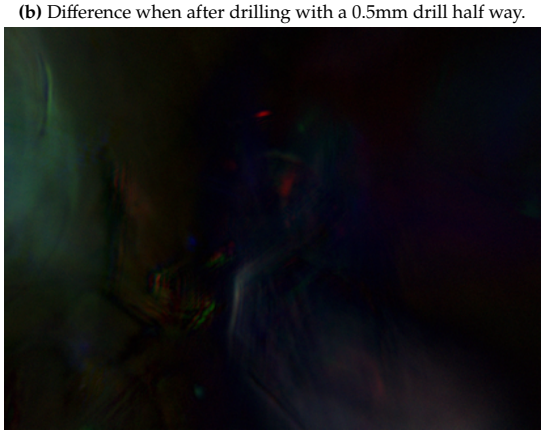


(a) Difference when after drilling with a 0.1mm drill.

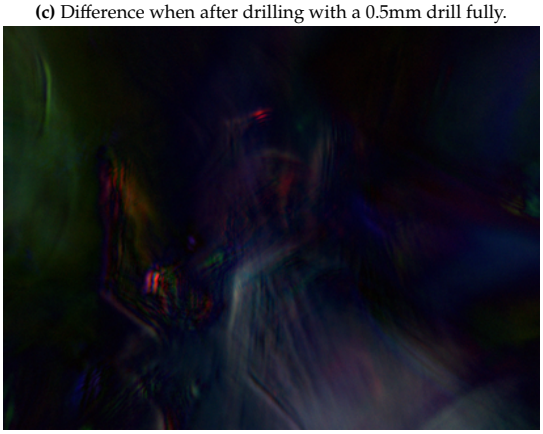
(b) Difference when after drilling with a 0.5mm drill half way.



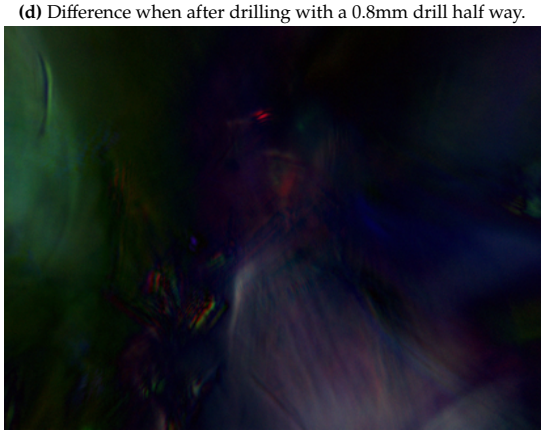
(c) Difference when after drilling with a 0.5mm drill fully.



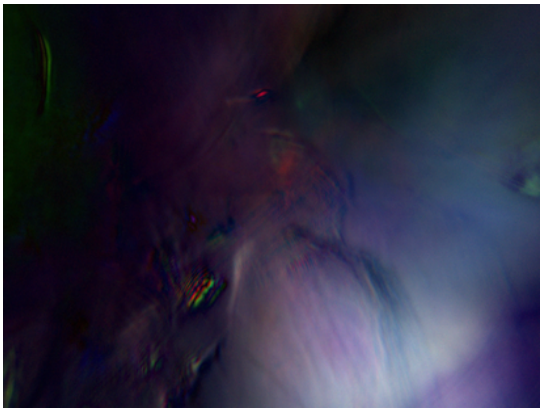
(d) Difference when after drilling with a 0.8mm drill half way.



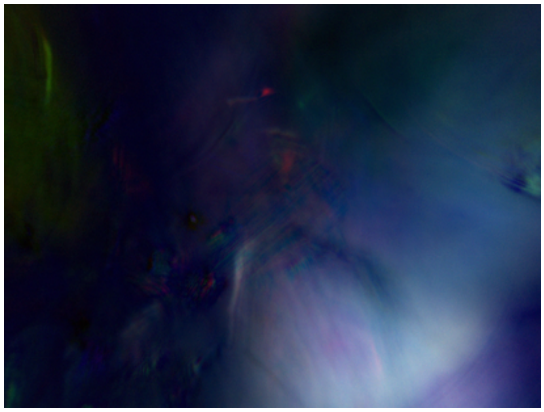
(e) Difference when after drilling with a 0.8mm drill fully.



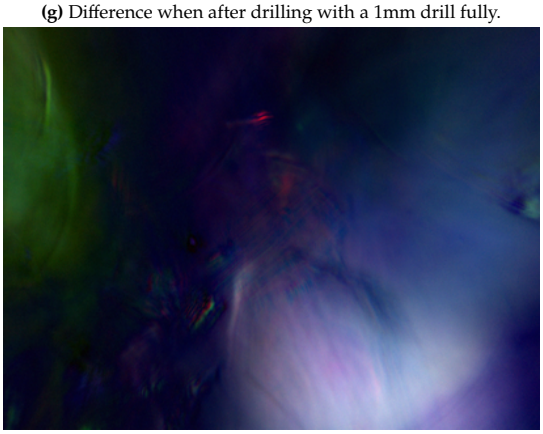
(f) Difference when after drilling with a 1mm drill half way.



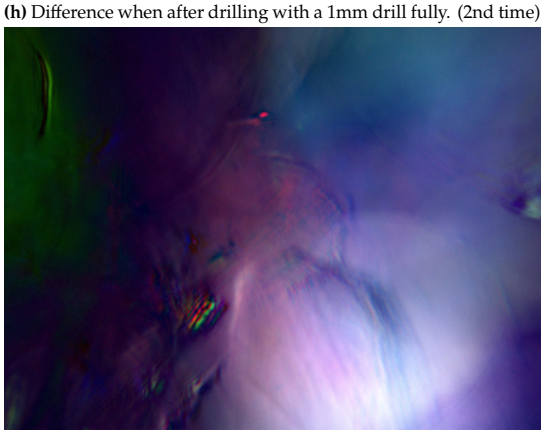
(g) Difference when after drilling with a 1mm drill fully.



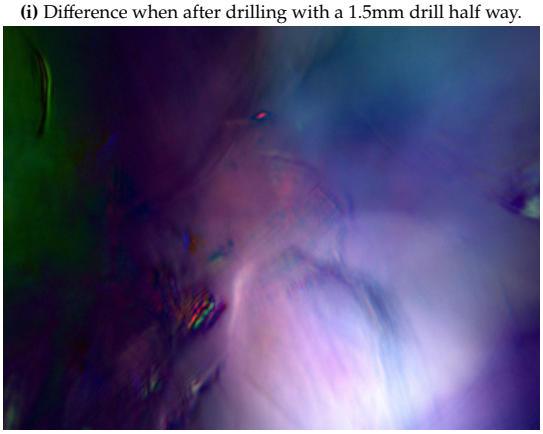
(h) Difference when after drilling with a 1mm drill fully. (2nd time)



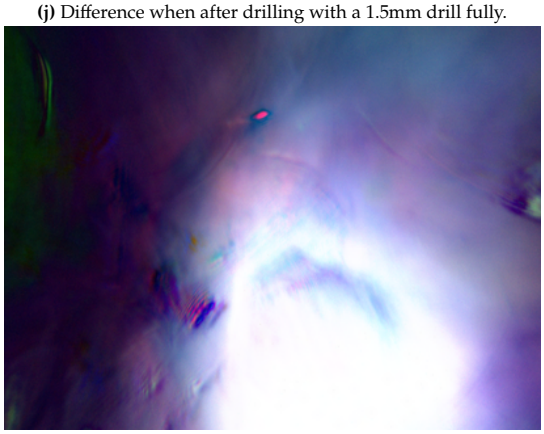
(i) Difference when after drilling with a 1.5mm drill half way.



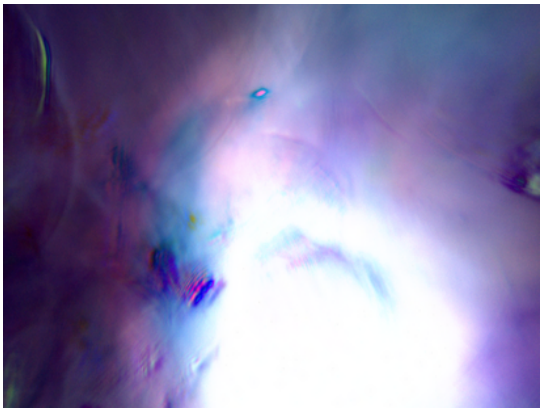
(j) Difference when after drilling with a 1.5mm drill fully.



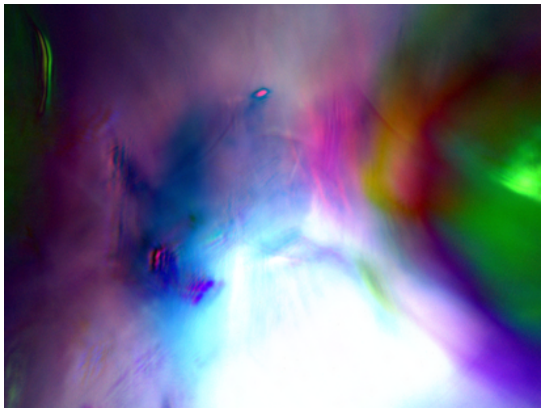
(k) Difference when after drilling with a 2mm drill half way.



(l) Difference when after drilling with a 2mm drill fully.



(m) Difference when after drilling with a 3mm drill half way.



(n) Difference when after drilling with a 3mm drill fully.

Figure D.0.1: Differences (scaled 25 times) in the drill test of the mean of the picture settings.

E

Relevant Data of the Picture Used in the PUF

Full Coloured
Given an full colour image with 3, 8 bit colour channels. Being 100% of the data. Then when bit-shifting by two will remove: $(8 - 2 = 6, 6/8 = 0.75) - > 25\%$ Then when only keeping 50% of the data will yield: 75%/2 = 37.5% So the coloured image with 50% of the data and a bit-shift of 2 will have 37.5% of the data.

Grey Adaptive Equalized
Given a grey-scaled image with 1, 8 bit channel. Being 100% of the date. than compared to a full coloured image will be 33.33% of the date. Bit shifting by two will remove an additional 25%. Then when only keeping 20% of the data will yield: $0.3333 * 0.75 * 0.20 = 5\%$ So the grey-scaled equalized adaptive image with 20% of the data and a bit-shift of 2 will only have 5% of the data. Compared to the colourd image (37.5/5 = 7.5) the grey adaptive is 7.5 times smaller. Or 5/37.5 = 13.333%, the grey adaptive image is only 13.33% of that of the coloured one.

F

Visualization of Drill Effect

Figure F.0.1 displays the visual effect after a 2mm hole has been drilled. This picture is made by comparing Figures 3.7.3a and 3.7.3b. Here it can be seen that the differences introduced by drilling is done so on the stable part of the pixels.

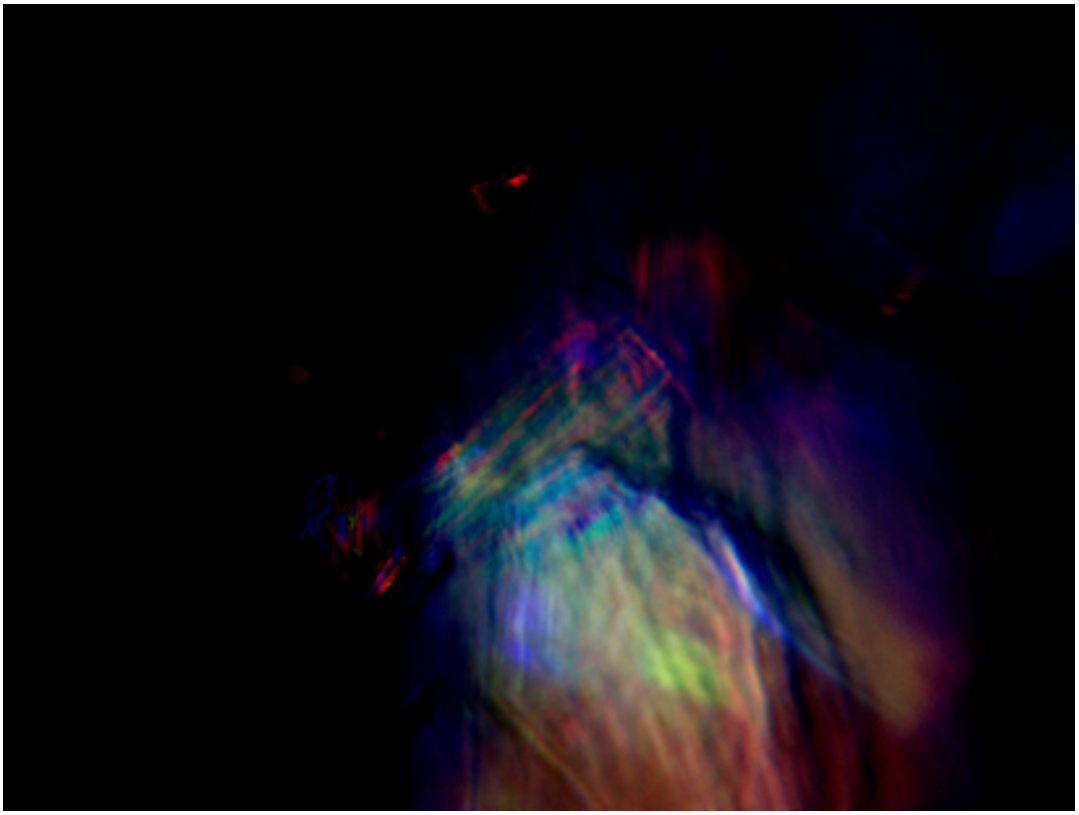


Figure F.0.1: Picture showing the stable pixels that change when being drilled with a 2mm drill.