

# PRESERVING FRESHNESS AND CONTINUITY IN REMOTE BIOMETRIC AUTHENTICATION

**ANKIT SARKAR**  
(B.Tech Computer Science, 2012)

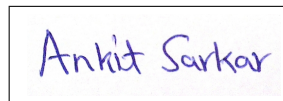
THESIS SUBMITTED FOR THE DEGREE OF  
MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE  
SCHOOL OF COMPUTING  
NATIONAL UNIVERSITY OF SINGAPORE  
AUGUST 2016

# Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.



Ankit Sarkar

4 August 2016

# Acknowledgements

Firstly I would like to express my gratitude to Associate Professor Chang Ee-Chien, for his guidance and encouragement. I would like to thank Professor Mohan S. Kankanhalli, Dr. Lekha Chaisorn, Associate Professor Terence Sim and Associate Professor Roger Zimmermann for their helpful suggestions about my research.

My sincere thanks to everyone at Sesame - Prabhu, Padmanabha, YongKang, Francis, Hung, Xiaolu, Erick, Adele and Thayalini. I shall always remember your help not only in research but also in my life.

Thank you all my friends at i.CARE - Benjamin, Cheryl, Hyqel, Egor, Richen, Priscilla, Prab and numerous others. I had a lot of fun with you guys!

Finally I would like to thank my parents and my younger brother who have always been there for me and encouraged me in my life. I could not have done this without you.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Objectives and contributions . . . . .	2
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	Authentication . . . . .	5
2.1.1	Authentication Factors . . . . .	6
2.1.2	Freshness in Authentication and Challenge Response . . . . .	9
<b>3</b>	<b>Review of state of the art</b>	<b>11</b>
3.1	Continuous Authentication . . . . .	11
3.1.1	Biometrics and Continuous Authentication . . . . .	12
3.1.2	Biometric Modalities used in Continuous Authentication . . . . .	12
3.1.3	Possible attacks . . . . .	14
3.2	Challenge Response . . . . .	16
3.2.1	CAPTCHA . . . . .	18
3.2.2	Biometric Fusion based approaches . . . . .	19
3.2.3	Reflection based Challenge Response . . . . .	19
3.2.4	Potential for research . . . . .	20
3.3	Video Tampering Detection . . . . .	20
3.3.1	Types of Forgery Detection Techniques . . . . .	22
<b>4</b>	<b>Proposed system design</b>	<b>27</b>
4.1	Challenges . . . . .	27
4.2	Proposed Model . . . . .	28
4.2.1	Verifier (Initial Authenticator) . . . . .	29

4.2.2	Presence Detector . . . . .	30
4.2.3	Video Tampering Detector . . . . .	30
4.2.4	Challenge Response verifier . . . . .	31
4.2.5	Integrator . . . . .	32
4.3	Proof-of-concept implementation . . . . .	33
4.3.1	Verifier . . . . .	34
4.3.2	Presence Detector . . . . .	34
4.3.3	Challenge Response . . . . .	36
4.3.4	Video Tampering Detection . . . . .	38
4.3.5	Overall System Score . . . . .	40
<b>5</b>	<b>Evaluation and discussion</b>	<b>41</b>
5.1	Experiments on Challenge Response . . . . .	41
5.1.1	Time to solve . . . . .	41
5.1.2	Discussion . . . . .	43
5.2	Experiments on Presence Detector . . . . .	44
5.2.1	Discussion . . . . .	45
5.3	Experiments on Video Tampering Detector . . . . .	45
5.3.1	Frame insertion and deletion . . . . .	45
5.4	Frame freezing . . . . .	46
5.5	Alternative forward-reverse looping . . . . .	46
5.6	Discussion . . . . .	46
5.7	Combined experiment . . . . .	47
<b>6</b>	<b>Conclusion and future work</b>	<b>49</b>
6.1	Future Work . . . . .	49
6.2	Open problems . . . . .	50

# List of Figures

3.1	Attacks on a biometric system. Adapted from [33] . . . . .	15
4.1	Proposed System Design . . . . .	29
4.2	Screenshots of the presence detector showing the face tracking on the left and the t-shirt area on the right . . . . .	35
4.3	Screenshot of a successfully solved challenge response . . . . .	37
5.1	Histogram of time taken by participants to solve the challenge response . . . . .	42

# List of Tables

3.1	Comparison of continuous authentication systems based on ability to check freshness of authentication information . . . . .	17
5.1	Table showing accuracy of video tampering detector in frame insertion and deletion . . . . .	46

# Abstract

Most remote authentication schemes perform authentication at the beginning of a session but have no way of ensuring whether the current user is still the originally authorised user. Continuous biometric authentication solves this problem by checking the identity of the user continuously. Despite the advances in this field, existing continuous authentication systems remain vulnerable to replay attacks as they do not have mechanisms to check the freshness of the authentication information. We survey the current state of art, compare existing systems and describe their vulnerabilities as well as identify the challenges. Based on these, we propose a model which ensures the freshness and integrity of the authentication data and prevents replay attacks. In addition, we also propose a new challenge response scheme for continuous authentication. We implement a proof-of-concept system and show that it is able to preserve freshness without significantly compromising on usability. Finally we suggest future directions for research.



# Chapter 1

## Introduction

### 1.1 Introduction

Most computer systems which provide services for a particular set of users, have a method to identify if a user is indeed the same person she claims to be. This serves to ensure that only an authorised user can access the services. Thus, authentication is an extremely important step to ensure access control. Over the years, many authentication schemes have been proposed [8] to protect a system from being accessed by unauthorised users. One of the most common ones in use today is the secret password which is used widely in email accounts. Other systems which are more sensitive might use biometric information like fingerprints to authenticate. Work is continuously being done to make authentication system resistant to attacks.

There is however one way in which an attacker could get access to the system. Consider the case of an employee who has just logged in to her email account. We assume that the account uses a password and secret one time code sent to the user's mobile phone for authentication. This system looks reasonably secure. A passerby with malicious intent will not be able to log in to the email account as he neither knows the password nor possesses the employee's mobile phone. So far so good. Now suppose the employee logs in to her email account and starts working. After a while she goes away leaving her account open. The passerby seizes this opportunity to look through the email and possibly grab hold of some sensitive information. The scenario described above is not impossible and neither

does it require a lot of skill. It can work even if the authentication system is highly secure.

The problem in the above example is that the email account only authenticates at the beginning of the session and then trusts the user till the session expires. It has no way of understanding that the person now using the account is not the person who was originally using it at the time of authentication. To prevent this, there is a requirement for authentication to happen continuously all the time. This is known as continuous authentication. In a continuous authentication system, the identity of the user is verified continuously which makes it hard for an attacker to break into the system. Over the years quite a few continuous authentication systems have been proposed using various biometric modalities like keystroke data [29, 38], face recognition [24, 31, 21], fingerprint recognition [39, 49], touchscreen behaviour [12] etc.

A continuous authentication system must be robust to attacks. Most previous works in this field have considered “zero-effort” [19] attacks on continuous authentication systems where the biometric traits of an intruder might be similar to that of a genuine user. Research has primarily focused on reducing the False Accept Rate to minimize the probability of a zero-effort attack. However, as Jain et al. [19] pointed out, biometric authentication systems are also vulnerable to “adversary attacks” where a determined attacker could impersonate a genuine user by using “a physical or digital artifact of a genuine user” or even by manipulating her own biometric trait. These can include compromising the sensor itself [24] and tampering the actual biometric data used for authentication. Despite this vulnerability, existing work has generally assumed that the sensors can be trusted. An ability to tamper with the authentication data can lead to replay attacks where an attacker digitally replays a previously recorded biometric information.

## 1.2 Objectives and contributions

In our work, we survey existing continuous biometric authentication systems and examine their vulnerabilities. In particular, we focus on the replay attack.

We show that a replay attack occurs due the fact that the systems do not have any mechanism to check the freshness of the authentication data. Our main contribution is the proposal of a model which we believe is resistant to replay attacks while being usable. We also propose a new challenge response scheme which can be used for continuous authentication over video. Finally we evaluate the proof-of-concept implementation and discuss future challenges.

The rest of the paper is organized as follows. In Section 2, we list out the basic concepts and types of authentication. In Section 3, we summarize the continuous authentication systems described in literature and describe their modalities used, the system design and vulnerabilities of these systems. We also examine relevant previous work in challenge response and video tampering detection. In Section 4, we describe our proposed system and our proof-of-concept implementation. In Section 5, we discuss our experiments and evaluation. Section 6, concludes our paper and points out the future direction for this research.

This page intentionally left blank

## Chapter 2

# Background

### 2.1 Authentication

Authentication is defined as “*verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system*” [32]. For the purpose of access control, a record of particular users (who are allowed) is maintained by the system. We refer to these users as authorised users as they are authorised by the system. All other users are termed as unauthorised users. When an unknown user (that is a user whose identity is not known to the system) requests access to a system by announcing its identity, the system must first verify if it is indeed the user it claims to be and then allow it access if the particular user name is on the authorised user list. In this context, it is the job of the authentication mechanism to verify that the unknown user is indeed the user it claims to be. This is a crucial step in any security system as it prevents an imposter from masquerading as an authorised user in order to gain access to the system. Thus, a failure in the authentication mechanism can compromise the security of the system.

In the case of user authentication mechanisms, failure can usually result in two types scenarios -

1. False Rejection - In this case the authentication system falsely rejects authorised users and does not allow them to access the system.
2. False Acceptance - In this case the authentication mechanism falsely allows

unauthorised users to access the system.

Authentication can be performed in multiple ways. An email might require the user to enter a password. An ATM machine would ask the user for the ATM card as well as a secret PIN before a transaction can be carried out. Immigration gates at certain airports requires an individual's passport as well as fingerprint to allow entry. Therefore, there are multiple methods of authentication which are in use today. We discuss some of the relevant concepts related to authentication in this section.

### 2.1.1 Authentication Factors

User authentication mechanisms rely on the fact that the user to be authenticated has some unique characteristic or information or possession. These are also known as the authentication factors [4]. We briefly describe these below.

1. **Something you know** - This could be a password, phrase, answer to secret question or any other knowledge which is known both by the user and the authentication mechanism verifying the user. The most common example is the password associated with user accounts and the PIN associated with bank cards.

This kind of authentication is widely used because of its convenience. The user just needs to memorize a particular fact and recall it while accessing the system. In the case of a password, the user needs to recall just one word. However, there is one big disadvantage due to the fact that knowledge can be transferred from one person to another. If a genuine user shares her password with another user, that user can use it to log in as the original genuine user. If a malicious user can get hold of a genuine user's password, she can use it to fool the authentication mechanism and consequently impersonate the genuine user.

Authentication mechanisms based on "something you knows" are thus vulnerable to attackers who have stolen the authentication information. To prevent this, users are advised to keep their passwords (or other identity information) secret. Despite this vulnerability, this type of authentication

is preferred due to its usability.

2. **Something you have** - This refers to something which only the user has with her in person. This must be something physical carried by the user. For example, when a user uses her debit card for shopping, the presence of the card helps to authenticate her. Another example could be the USB device used to verify the licensed users of a software. The software could be installed on a shared machine, but the licensed users have a USB device which can be used to enable the software.

Possessing a physical object for authentication offers a significant advantage over remembering a piece of information. Knowledge is usually stored digitally. This makes it easier for a malicious user to copy the knowledge and use it. The attacker can carry out her attack remotely and know the shared secret, In this case however the attacker would need to gain physical access to the device. This makes it a lot harder to carry out attacks.

Physical objects can however be stolen which presents a weakness. A malicious user possessing a genuine user's physical identifier can essentially identify as a genuine user. Credit card frauds (in which a stolen credit card is swiped to pay for unauthorised transactions from the genuine user's account) are a longtime problem.

3. **Something you are** - This refers to something which is inherent to a person. In other words, it means a set of characteristics like face structure, voice or behaviour which are naturally present in the person. These characteristics must be unique to differentiate between multiple users. One way to quantize these characteristics is to use biometrics.

Authentication based on biometrics [20] offers significant advantages over those based on knowledge and possession factors. For a start, they are hard to copy by another person. For example, in the case of face recognition, disguising the face to impersonate another user is hard. In addition, the user does not need to explicitly remember something (knowledge) or have a physical object (possession). This removes the risk of attacks due to theft of the knowledge or possession. However, it must be noted that biometric

security systems are not foolproof [33]. In particular, once an attacker gets hold of the original biometric information (fingerprint design or video of face image), she can replay it to gain access to the system. To prevent this, the system needs to detect replay attempts.

Based on the use of the above mentioned factors, authentication systems can be classified into two main types.

1. **Single Factor Authentication** - Single factor authentication relies on using only a single type of factor for authentication. The most common example is the conventional email account which is protected by a password. Another example could be the use of fingerprint recognition for access control.

Single factor authentication are simple and easily implementable as well as usable. They are typically used in scenarios where the impact due to a security breach will not be substantial and using a sophisticated security system might incur a lot of overheads. However, they are vulnerable to attacks since it is relatively easy to compromise a single factor.

2. **Multifactor Authentication** - Multifactor authentication relies on using two or three different types of authentication factors. A common example could be the use of an ATM card to withdraw money. The ATM card is something the user has (possesses) while the PIN used is something the user known. Immigration gates at some airports require the person to scan the passport (something the user has) and the fingerprint (something the user is) for authentication.

Multifactor authentication is considered more secure than single factor authentication [22]. This is because even if an attacker successfully replicates one factor, it would be difficult to replicate the other factor. However, in terms of usability, it may not be as user friendly as single factor systems. It might also take more time to process two factors. As a result, it is usually used in sensitive applications like online banking. In such cases, an enhanced level of security is more important as compared to the usability of the system.



### 2.1.2 Freshness in Authentication and Challenge Response

Freshness [8] is a concept linked to the notion of time in authentication. When a user transmits a secret (suppose it is a password) to the system for authentication, it is vulnerable to eavesdropping. A malicious user after acquiring the secret, could later “replay” it and impersonate the genuine user. The vulnerability remains as long as the authentication system uses the same secret multiple times for identity verification. Thus, as stated by Lam et al. [27], there is a need to “*provide freshness assurance of objects for applications that are vulnerable to replay attacks*”.

One way of achieving this is to use a Challenge Response mechanism [27]. In a challenge response mechanism, the system sends a random number (challenge) to the user who must then incorporate this challenge in her message (response) to the system. If the challenges produced each time are unique, a replay attack will not work. This is because a previously recorded response will contain only the answer to a previous challenge. To strengthen the scheme, it is recommended to allow the challenges to be readable only by the user [27]. If the challenge is open, then the scheme must provide a guarantee that the response has come from the legitimate user.

This page intentionally left blank

## Chapter 3

# Review of state of the art

### 3.1 Continuous Authentication

Continuous authentication refers to a system which continuously verifies the identity of the user throughout the session. When a user requests access to the system, an authentication process is done which is similar to conventional authentication (by conventional we mean authentication which only verifies the user only once at the start of a session) and the user is allowed access to the system. In a continuous authentication system however, the authentication process continues in a loop [39, 26] and keeps verifying the user, even after the user has been granted access. This provides a guarantee (at every instant of time) that the same user who logged in at the start of the session is the one who is still using the system. Continuous authentication systems, thus, avoid the type of attacks in which an attacker can hijack the authenticated session. If the original user leaves the system unattended and another user tries to use it, the system can block access immediately.

Continuous authentication can be used in scenarios when preserving the security of the system is of utmost importance. One particular application could be in the cockpit of an aeroplane [5]. The system can be configured to allow only the authenticated pilots control the movement of the plane. This property could be useful in preventing an intruder from taking control of the plane in case of a cockpit breach. A continuous authentication system could also alarm the ground control to a potential aircraft hijacking. Another application could be

while conducting online examinations [11]. While MOOCs (massive open online courses) have increased in popularity in recent years, one of the limitations such courses have is that there exists no foolproof way to conduct a remote examination. In a conventional examination, an invigilator ensures that examinees do not leave the area and someone else doesn't take the exam on behalf of the examinee. In a remote examination, this can be ensured by using continuous authentication. The continuous authentication system could track the examinee continuously and raise warnings in case the examinee attempts to leave the room or someone else is also detected in the field of vision.

### **3.1.1 Biometrics and Continuous Authentication**

Biometrics refer to certain unique characteristics which are inherently present (“what you are”) in a human. These characteristics can be either physiological [28] which includes physical features like face, fingerprint, iris etc. or behavioural which refers to certain ways of performing actions like walking(gait), typing (keystroke dynamics) and writing(stylometry) etc. Using biometrics in continuous authentication systems offers usability significant advantages. Continuous authentication based on “something you know” (for example, a password) would require a user to constantly input the identification information to the system [31]. If authentication is based “something you have”, it would require the user to constantly carry the object for authentication. In contrast, physiological and behavioural characteristics are inherent in a person (no need to explicitly carry it). Depending on the biometric modality and the sensor used, this can be passively recorded without requiring the user's cooperation. Thus, biometrics offers a significant practical advantage over other methods of authentication.

### **3.1.2 Biometric Modalities used in Continuous Authentication**

Biometric continuous authentication systems using various modalities have been explored in existing literature. One of the early works used keystroke dynamics to perform continuous authentication [38, 29]. This was advantageous as it it could be run passively in the background without the need for user cooperation. Another work proposed a framework for using continuous authentication in an

aircraft cockpit [5]. Further work explored the use of other modalities like faces, fingerprint, mouse and touch dynamics for recognition. However, using only a single modality makes the system vulnerable to momentary failures [2] due to change in external factors like lighting, noise etc. More recent work has focused on fusing input from multiple modalities to enhance the system [2, 39, 49]. In response to the fact that conventional biometric authentication are computationally expensive, soft biometric features like the colour of the user's clothes [31, 30] have also been used. We briefly describe the most frequently used modalities for continuous authentication:

- **Keystroke Dynamics:** Keystroke dynamics refers to the way in which a user types. It is a kind of behavioural biometrics and some of the earliest works in continuous authentication have focused on using it [38, 29, 13]. One of the main advantages of this scheme is that it does not require active cooperation of the user and can be run passively. It also doesn't use any specialised equipment. However, the downside is that it takes some time before the data is enough to authenticate a user and by this time, an attacker could do some damage to the system. Recent work [34] has explored the keystroke sounds instead of the keystroke timings as a biometric identifier.
- **Mouse Dynamics:** Mouse dynamics are another kind of behavioural biometrics which can be used to authenticate users [23]. Similar to keystroke dynamics, mouse dynamics offers the benefits of passive observation and no need for specialised equipment. However, previous work on mouse movement requires significant amount of initial data to make a decision [23] and have not shown enough accuracy to be accepted as reliable for a continuous authentication system [41].
- **Touchscreen Behaviour:** Touchscreen use has been increasing due to the surge in popularity of smartphones and tablets. A few works have demonstrated that touchscreen behaviour does have potential to differentiate between users and can be used for continuous authentication [12]. However, at the moment there is a lot of room for improvement in the

accuracy of the system [9]. In addition it requires a large amount of training data [48]. Thus, while it is not very useful in the present, it is worth exploring for the future.

- **Face:** Face recognition is a popular way to perform continuous authentication [24, 31, 21]. As a biometric modality, face recognition is reasonably accurate and can be performed passively (without active user cooperation) using just a camera. In addition, unlike behavioural biometrics like keystroke recognition, face recognition does not need to wait for a certain amount of time in order to have a minimum number of samples [36]. This makes it a popular modality for continuous authentication systems which need to balance the need for security with usability and time.
- **Fingerprint:** Fingerprints have been proven to provide a high level of accuracy and can theoretically be used for continuous authentication. However, it is usually not a convenient modality to use because capturing of fingerprint requires user cooperation [36]. It can however be used as an additional authentication modality when fusion of multiple modalities is done [39, 49].
- **Soft Biometrics:** Soft biometrics are defined as “characteristics that provide some information about the individual, but lack the distinctiveness and permanence to sufficiently differentiate any two individuals” [18]. Some of them are gender, ethnicity and colour of eyes, skin, hair or clothing. While they cannot be used alone for authentication, they can be used in combination with hard biometrics and help to enhance the security of the system. Niinuma et al. [31] have shown that soft biometrics can be used in continuous systems since they remain permanent at least for the duration of the session and can help to track the user in cases where a conventional biometric feature, such as a face, cannot be detected.

### 3.1.3 Possible attacks

A biometric authentication system, like any authentication system, is vulnerable to attacks. Certain types of biometric information can be easy to steal such as a

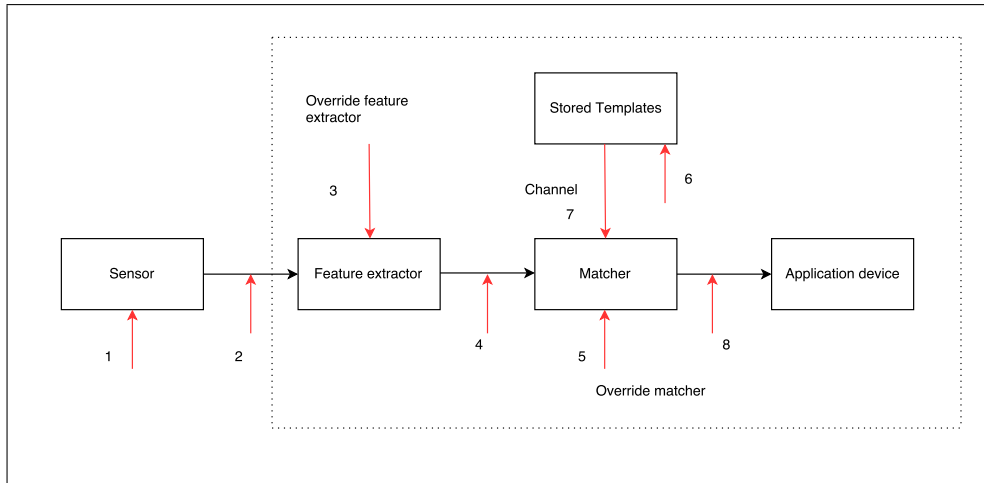


Figure 3.1: Attacks on a biometric system. Adapted from [33]

face image or fingerprint. Consider a system which uses face recognition to allow access. An attacker can obtain the face image of the genuine user (for example, surreptitiously clicking a photo or stealing one from social media), print it out, wave it in front of the camera used by the system and thus, fool it into allowing access. A more sophisticated attack could be to compromise the network and inject a digital copy of the face image directly into the verification system. These two examples show that biometric information can be easily stolen and reused. Thus, to quote to [35], for a biometric system to work well, it must be able to verify two things: (a) *that the biometric came from the person at the time of verification*, and (b), *that the biometric matches the master biometric on file*. While much work has been done to improve biometric matching, comparatively less work has been done to address the first point. Attacks of biometric systems have been classified into 8 possible types depending on the point of attack (see figure 3.1) [33]. In the specific case of a remote biometric system, points 3-8 would require an attacker to successfully attack the backend server. While such attacks are possible, it is also possible to prevent these by taking steps (such as a firewall and intrusion detection system) to protect the server. On the other hand, points 1 and 2 lie on the client side which the attacker can more easily exploit. To attack point 1, the attacker would need to fool the sensor (using fake biometrics) that live biometric information is being input. To attack point 2, the attacker can digitally replay previously recorded information. The former is known as a Spoofing attack while the latter is called a Replay attack.

Spoofing attacks refer to attacks where a fake biometrics is presented to the sensor. A possible reproduction of a biometric modality is made and presented to the system. These could be a fake finger, a photo of a face, a face mask, Spoofing attacks have been well studied in literature.

Replay attacks refer to scenarios where the biometric information is digitally replayed, bypassing the sensor. This is harder to detect. Most previous approaches to solve this have involved using a trusted sensor which can integrate time information into the biometric information. Replay attacks on continuous biometric authentication systems have been comparatively less studied.

In table ( 3.1), we have looked at some major continuous biometric authentication systems (which passively obtain data from a biometric modality) and tried to see if they contain mechanisms to ensure freshness of data in order to prevent replay attacks. To our surprise we found that most such systems do not contain any such mechanisms. The only one in which replay attacks was discussed relied on using a trusted sensor. Unfortunately, in a remote authentication environment, that cannot be guaranteed. Thus, there is a need for a mechanism to prevent replay attacks. The ideal solution would be to have a passive mechanism to ensure freshness, without requiring a trusted sensor. In the absence of that, an active mechanism to ensure freshness would be good as well.

## **3.2 Challenge Response**

There is a need to ensure that the data used for authentication is fresh - that is, the data has been acquired “now” and is not old data that is just being replayed. If freshness is not ensured, an attacker could simply replay the old data digitally and gain access to the system. In a remote authentication system where the sensor may be unsupervised, it may be all the easier for an attacker to use this attack. Hence, a continuous authentication system must guard itself against a replay attack.

Measures to protect against replay attacks have been suggested over the years. Some of them involve the use of specially designed sensors. However, it



Table 3.1: Comparison of continuous authentication systems based on ability to check freshness of authentication information

Work	Modality	Checks freshness?	User cooperation required?
[38] , [29]	Keystrokes	No	NA
[34]	Keystroke sound	No	NA
[23], [41]	Mouse dynamics (Passively recorded, but not as accurate as other modalities)	No	NA
[24]	Face (passively recorded)	Yes, trusted camera	NA
[39],[49]	Multimodal (passively recorded though it requires specialised hardware)	No	NA
[18], [31]	Soft biometrics, passively recorded	No	NA
Ideal solution	Any passive biometrics	Yes, should be able to passively check.	No
Our solution	Face	Yes, but requires active human cooperation	Yes

it not practical to assume that. As such we do not survey these methods since we cannot assume that the sensor will not be compromised.

To ensure freshness, a challenge response scheme might be used. The authentication system must send a random challenge to the user and the user needs to respond with the correct answer. An ideal challenge response scheme should pose a different challenge every time so that it is not possible to use a previously stored answer. In addition, it should be able to quickly verify the response. In the case of biometrics, depending on the how it is implemented, a challenge response scheme should ideally require as less human cooperation as possible.

### 3.2.1 CAPTCHA

A relevant example of a challenge response scheme is a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) which is used to differentiate between computers and humans [43]. To differentiate between a computer and a human, the authenticator send a challenge to the remote unknown user. This challenge is typically a hard AI problem which cannot be solved by a machine. If a correct response is received, it can be inferred that the user is a human and not a bot. Despite the fact that the goal of a CAPTCHA is slightly different, we still find it relevant because the response to a CAPTCHA must be solved at that particular time and hence can help to ensure freshness. The earliest proposed CAPTCHAs relied on the fact that automatic algorithms could not read distorted text while humans could [43]. Since then a variety of CAPTCHAs have been proposed which rely on challenges based on image classification [10], image orientation [15], face recognition [16], audio recognition [17] and video understanding [25]. The response to these challenges are usually given in the form of a text input, dragging, and clicking/touching.

One particular property we notice is that almost all CAPTCHAs require active human cooperation to work. Typing, clicking and dragging require significant human cooperation and if invoked regularly, could potentially disrupt the user from using the system. Indeed, usability issues related to CAPTCHA have been well studied. However, a fully passive alternative has not yet been developed which could replace CAPTCHA entirely.

Another major disadvantage of a CAPTCHA is the fact that the response is usually a text, click or drag. This response is sent separately from the biometric data, something that is termed as out of band. For example, the live stream of a face video and a response text to a CAPTCHA can be sent from the same computer, but they are sent differently. On the server side however, it may be difficult to verify that both have come from the same place. There could be a possibility that a separate entity is solving the CAPTCHA while playing back the genuine user's video.

### **3.2.2 Biometric Fusion based approaches**

Biometric fusion based approaches are another class of approaches to ensure freshness and liveness [42, 7]. A typical challenge would be to perform a certain action and input from two or more biometric modalities would be analysed as the response. Based on the challenge, if there appears to be no synchronization between the two modalities, it can be assumed that the data is not fresh. This technique primarily uses audio as well as video response. The user is given a particular text to read out aloud. The voice sent as a response is then checked to see if the text has been spoken. Alongside, the movement of the face, particularly the lips is analysed to see the correlation with the voice.

This method is a good way to ensure freshness as well as liveness. It is difficult to simultaneously do a replay attack on two modalities, particularly when a random challenge is posed. However, it still suffers from the disadvantage that it requires active human cooperation.

### **3.2.3 Reflection based Challenge Response**

A new mechanism proposed recently uses a concept analogous to watermarking for ensuring freshness [40]. The technique relies on the fact that when a user is using a computer/mobile device, the light from the screen is reflected off the user's face, particularly the eyes. This light can then be continuously analysed. The challenge in this case could be a change in the intensity or colour of light and the response would be sent as part of the video stream recording the face. A huge advantage of this approach is the fact that it does not require user cooperation

an neither any special hardware. The process can continue without disrupting the user. In addition, the information ensuring freshness is sent in the video signal itself and not separately. This is advantageous as it presents less avenues for modifying the authentication data.

However, the work has only been tested in a darkened environment and the authors have mentioned that the light reflected from the device screens is quite less. Thus, it remains to be seen how well it can perform under varying lighting conditions and different kinds of screens.

### **3.2.4 Potential for research**

From the above survey it is clear that there is scope for more research in freshness detection techniques. Due to reasons mentioned above, CAPTCHAs are not fully suitable for our remote authentication system. A multimodal challenge response will work, but it will also require the user to actively cooperate. In contrast, the reflection based mechanism can be used passively without user cooperation. However, to the best of our knowledge, this is the only passive challenge response mechanism we have come across which is applicable for face recognition systems. Although definitely a good start it is currently not known how this approach will work under varied lighting conditions, cameras and device screens. Thus, there is a need for a challenge response system which is passive,

One possible future work in this area could be to improve the reflection based challenge response so that it works under different lighting conditions. The approach needs to be made more generalised to work for different screens and cameras. The colour based approach can also be improved and it would be interesting to study the reflections on a user's face with changing patterns.

## **3.3 Video Tampering Detection**

In recent years, video based surveillance methods have become increasingly popular. Banks, offices, schools and even homes are being equipped with CCTVs. Video surveillance offers two key advantages. One is the ability to watch over multiple places without being physically present. A security offer could sit in

his office and watch the feed from multiple cameras at the same time. These camera are remotely connected to the office server and broadcast live feed from the area, thus, reducing the need for law enforcement officers to be present at all places. The second advantage of video surveillance is for investigative purposes. In a post crime scenario, video footage can offer vital clues about the crime. However, all this is valid only on the assumption that the video stream has not been tampered with. If the video stream has been modified, it can be possible to hide certain details which can mislead an investigation which in turn can have far reaching legal consequences. Thus, it is important to get an assurance that the video in question is genuine and not tampered in any way.

Since a video is considered as a set of images played at a particular frequency, the same methods which are used for image tampering can also be used for video tampering. Broadly speaking, video tampering can be considered to be of two types intra-frame forgery and inter-frame forgery.

1. Intra-Frame Forgery refers to the modification of a particular frame. This is very similar to image forgery. In this kind of forgery, the attacker takes a frame, modifies certain regions of it and then re-encodes the video. It is important to note that the changes done are only to the content of the frame. The temporal properties of the frame, such as its placement with relation to other frames is not disturbed. An example of intra-frame forgery could be the replacement of a certain region in a frame with another region from the same or different frame. This could be applied to successive frame so that when the video is played, the original content of the regions is not visible.
2. Inter-Frame Forgery happens when the attacker changes the temporal properties of the frames in the video. This can be done by deleting, inserting or duplicating frames. As an example, an attacker could remove a set of frames containing a particular object. If this video is subsequently played, viewers would not be able to see the particular object. It should be noted that in this case, the content of the frame itself is not being modified. A replay attack as we have previously described is actually an instance of

inter-frame forgery.

In a real world scenario, it is quite possible that both types of forgeries are applied simultaneously on the same video. This can make it especially challenging to detect the modified frames and the regions where they have been modified. However, in this paper we limit our discussion only to Inter-Frame forgeries and assume that the content of the frames itself have not been tampered with.

### 3.3.1 Types of Forgery Detection Techniques

A video is always captured using a sensor and one of way of tampering the video is to compromise the sensor itself. This is not particularly difficult since many cameras are mounted on remote locations. To prevent this would require physically protecting the sensor which is not always feasible and defeats the purpose of having a remote surveillance camera. Another way to tamper the video stream could be to intercept the communication on the network. For this there are existing methods which encrypt the data being transferred although they may not always be foolproof. To prevent the tampering entirely can be hard to achieve. Hence, there need to be methods which can at least detect is an incident of tampering has taken place. Over the years various measures have been proposed to determine if a video in question has been tampered with or it. According to Chao et. al.[6], video forgery detection techniques can be classified into 2 types Active detection and Passive detection.

Active detection is analogous to the concept of digital signatures. A video is water marked and digitally signed beforehand using a secret and this can be verified later to ensure it is genuine. However, a significant disadvantage of this type of technique is the fact that it relies on a secret. If an attacker gets to know the secret, he can encode the same watermark and signature on the tampered video. In addition this scheme would also require the camera to have the functionality to watermark the video. Unfortunately many cameras do not have this inbuilt facility making it hard to use this technique in practice.

Passive detection techniques on the other hand focus on detecting forgery from the video itself. These techniques do not need to rely on any secret embedded into the original video. The basis of these techniques rely on the fact

that any insertion or deletion of frames will cause certain features of the video to change. These features can then be exploited to understand whether a tampering has occurred. The two main passive detection techniques we have surveyed rely on using artefacts from double compression and optical flow respectively. We describe these in the sections below.

### **Double Compression Based Forgery Detection**

A video consists of a many still images (frames) in a sequence which are played at a particular rate. However, such a sequence would otherwise occupy a large amount of space. Encoding algorithms (like MPEG-2) have been developed which can greatly reduce the size of video. These algorithms store videos as a collection of different kinds of frames. I frames which are used as reference points P frames uses data from previous frames B frames which can use data from previous as well as future frames.

P and B frames are predicted based on other frames. Since there is a lot of redundancy between frames, a frame can be expressed as another frame plus some residual error. This residual error can then be coded which takes up less space than the actual frame itself.

This process when applied over the video can lead to a significant reduction in size. However in practice the video is divided into segments and then the coding algorithm is applied. This is to prevent error incurred during decoding a frame to continue propagating to the next frames. Each such segment is referred to as a GOP (Group of Pictures). A GOP starts with an I-frame and is then followed by P-frames and B-frames. The I-frame itself is not predicted and is compressed as is.

Wang and Farid's work [46, 47] exploited this structure to detect if any inter-frame forgery has taken place. Consider the original video which is divided into GOPs. When a frame is deleted, the video needs to be recompressed for storage. However, due to the removed frames, the GOPs for the tampered video need to be reordered. Subsequent frames take the place of the missing frames in the previous GOP. On encoding it, the original I-frame of the GOP is double compressed while certain frames will move from one GOP to other. This gives

rise to certain static and temporal changes.

The static changes can be observed in the I-frame which was re-encoded. If the Fourier transform of the histograms of the I-frame is taken, spikes imply double compression. The temporal changes observed are due to the shifting of P-frames and B-frames. Compression of the original video has the effect of making frames within each GOP correlated to each other. When frames move to a different GOP and are re-encoded, these frames have higher estimation errors as compared to frames which were originally in the GOP.

The above method will however not work if an entire GOP is removed. Furthermore, the static artefacts assume that the quantization level is same within a frame. However, if the coding algorithm uses varying quantization levels even within a frame, this method does not work. Recent work [14] has attempted to increase the robustness of the approach so that it is not impacted by quantization levels or the location of the tampered frame. Furthermore similar techniques have been used to localise the point of frame insertion and deletion. This however also fails if an entire GOP is removed. Other works [37] in this area have focused on solving this problem when a constant bit rate is used for coding videos. This work uses a machine learning approach where untampered and tampered videos are used to train a classifier. The features used are based on a previous work and have been adapted for this scenario.

### **Visual Artefacts Based Forgery Detection**

Visual artefacts refer to the changes in visual properties of the video. This technique relies on the fact that any addition or deletion in frames would lead to abnormal changes in certain visual properties. Compared to double compression based forgery detection techniques, very few works has been done to explore visual anomalies due to tampering.

An early attempt [6] was made by representing videos using gray values and looking at the correlation between frames. This idea behind this approach is that any insertion or deletion in frame will result in a sudden change in the correlation coefficients of gray values. However, this work was tested on a database which contained videos taken with a still background. In real life this may not always



be the case, especially in videos where the foreground can contain a lot of moving artefacts while the background is usually occluded.

Recent work [45, 44] has focused on using Optical Flow to detect frame insertions and deletions. A video has a naturally occurring optical flow. If any forgery is done, it can introduce variations in the optical flow which is different from the naturally occurring flow of the video. The algorithm calculates the optical flow and the relative change in the optical flow is plotted. Based on the type of forgery (insertion or deletion) 1 or 2 discontinuity points are observed. Further work has attempted to use a Support Vector Machine to classify this.

Although this method can help to detect the type of forgery (insertion or deletion), the results may need to be revalidated. The authors have considered the insertion as a case where a frame from a different video is inserted. However, it is quite possible that a frame from the same video could be inserted as well. This is not addressed in the paper. In addition, the videos have all been taken from a stationary camera with no variation in background and it is unknown how the results will be when there is no stationary background.

This page intentionally left blank

## Chapter 4

# Proposed system design

### 4.1 Challenges

Jain et al. [20] have mentioned that a practical biometric system must take care of three issues - *performance*, *acceptability* and *circumvention*. According to their definition, performance refers to recognition accuracy and speed; acceptability refers to the ease of using the biometric characteristic; and circumvention refers to the ease with which the system could be attacked. We feel that the same applies to any biometric continuous authentication system, except that the requirement of continuous authentication makes it more challenging.

A continuous authentication system must have a high recognition accuracy and speed. Certain biometric recognition techniques can use up a lot of processing power. Using sophisticated attack detection techniques might prevent attacks but it could also slow down the system. The entire process needs to be made efficient enough for achieving a real time system.

We have noticed that it is possible for the sensor to be tampered with which offers attackers a lot of advantage. Unlike previously described systems which assume trusted sensors [24], we assume that the input sensor might be vulnerable and it is the job of the system to handle this problem. This is challenging because in addition to identification, the system now needs to actively defend against input tampering and spoofing attacks.

It has been shown in a previous work [19] that some biometric modalities are more reliable in differentiating between users than others. For example

fingerprint recognition is more reliable than keystroke dynamics. However in a continuous authentication setting, fingerprint recognition would need active cooperation of the user [36] while keystrokes could be observed passively. Therefore, the biometric modalities to be used need to be selected carefully. A continuous authentication system should ideally work passively without requiring active user cooperation. In case of challenge response, user cooperation might be unavoidable. Therefore, either the challenge response needs to be passive or the frequency of challenge response needs to be reduced. We realise that while our goal is to make the system increase the recognition accuracy, we cannot afford to compromise on usability.

Thus we see that system performance, attack resistance and usability are the three challenges we face while designing the system.

## 4.2 Proposed Model

Keeping the above in mind, we propose a slightly different approach. In our approach, biometric authentication (verification of identity) is done at the start of the session. Once the user has been successfully authenticated, the system would only need to ensure that the user is continuously present. In addition, the system should have a mechanism to detect any tampering of the authentication data as well as ensure that the data is fresh.

Suppose we are using face as our main biometric modality. The main advantage of this is that the face can be continuously recorded without requiring active user cooperation. Consider the scenario where the first  $n$  frames have been recorded ( $x_1$  to  $x_n$ ) and transmitted to the server (received as  $y_1$  to  $y_n$ ). Assume that these frames (content and order) have not been tampered with and are fresh. That is  $y_1 \equiv x_1$  and so on. We assume that a face recognition run on these frames verifies that the user present is the authentic user. Now suppose the camera records the next frame in the sequence  $x_{n+1}$  which is then sent across the network to the server. The server receives a frame  $y_{n+1}$ . Now in order to ensure that user is continuously authenticated for the  $n+1$ th frame as well (and the system is resistant to replay attacks), we need the following to be true

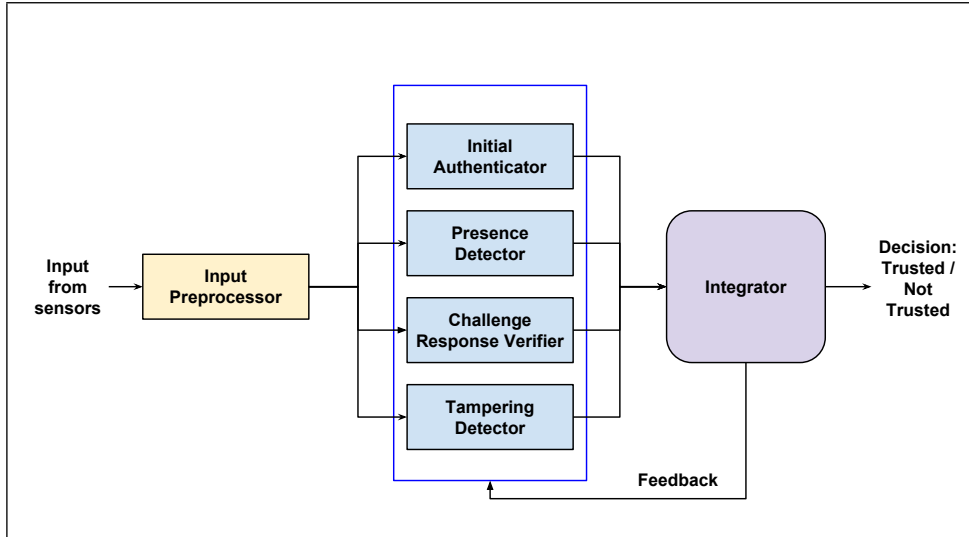


Figure 4.1: Proposed System Design

- $y_{n+1} \equiv x_{n+1}$  (That is, the frame has not been tampered in any way - contents or order)
- The user is present in the frame  $y_{n+1}$

We propose a system which checks for the above mentioned conditions and this ensures that not only is the user continuously authenticated, but also that the authentication data itself has not been tampered and that freshness of this data is assured. Our proposed system consists of four main components (see figure 4.1) - an initial verifier which verifies the identity of the user, a presence detector which continuously checks that the user is present, a video tampering detector which checks for possible tampering in video frames and a challenge response verifier which verifies that the user has correctly responded to the challenge in time. We describe each component below.

#### 4.2.1 Verifier (Initial Authenticator)

The verifier's main function is to verify the identity of the user. For this, the verifier can possibly use face recognition. Given an id and the user's face, the verifier checks if the presented face matches the one with the same id in the database. The verifier is usually run at the beginning of the session. As such, it is important that the verifier have a high accuracy rate so that an imposter is locked out at the beginning of the session itself. To improve the accuracy

of the verifier, multiple biometrics (even those that may require active human cooperation) may be used. As the verifier is run for a short duration at the beginning, it should not significantly impact the overall usability of the system.

#### **4.2.2 Presence Detector**

The main function of the presence detector is to check that the user present in the previous frame should also be present in the current frame. Assuming that the user present in the previous frame was the verified user, as long as the user doesn't leave the frame, it can be assumed that it is the same user. (Of course, there are ways to attack this - such as tampering the content of the video frame or inserting a frame from a different video, but a different component handles these attacks).

As the presence detector needs to run continuously, it should ideally not require any user cooperation. For this, tracking the face of the user is a good option. However, there may be moments when the face is not detected (such as a profile face or when the user bends down their head). It would be inconvenient if the user is logged out frequently even though they are present. To mitigate this, a second modality can also be used. This may be soft biometrics which remain unchanged throughout the session such as the colour of the shirt of the user.

In addition to the above, the presence detector needs to have a mechanism to check for biometric liveness detection. This is important to prevent spoofing attacks such as putting on a face mask resembling the face of the authenticated user.

#### **4.2.3 Video Tampering Detector**

The main function of the video tampering detector is to check that the frames recorded from the sensor on the user's side have not been tampered with. This involved checking for 2 conditions -

1. The contents of the frame itself have not been modified
2. The order of the frames have not been modified

For the first, intra-frame forgery detection techniques need to be used while for the latter, inter-frame forgery detection techniques are suitable. As our main focus is on replay attacks (inter-frame forgery), we henceforth, limit our discussion to inter-frame forgery.

As we have seen before, there are two different approaches for inter-frame forgery detection. The first, also known as active relies on using a secret embedded into the frame and then checking for. However, such an approach works on the assumption the sensor on the user's end can be trusted. As we cannot assume the same for remote authentication, we need to rely solely on the information in the frame itself. Accordingly, passive (or blind) techniques are more suitable here.

The video tampering detector needs to work passively in the background, similar to other components. The main approaches use compression artefacts or inconsistency in optical flow to detect possible tampering. The approaches based on compression artefacts however may require prior information about the type of compression and encoding used. For this reason, the topical flow approach, relying only on the content may be a better solution.

#### **4.2.4 Challenge Response verifier**

The main function of the challenge response verifier is to verify that the challenge presented to the user has been successfully completed within a given time. A completion of a challenge within a certain time threshold assures the freshness of the video stream - that is, the video is live and the frames have not been pre-recorded.

A necessary requirement of the challenge response system is that the response to the challenge must be sent "in band". That is, the response must be present inside the video frames itself. If this requirement is not fulfilled, it is not possible to assure the freshness of the video frame. For example, suppose the system which is continuously recording the face of the user, presents a challenge. The challenge response requires the user to read a distorted text and type the input. Note that the video is sent separately from the text data. An attacker can exploit this to replay a previously recorded video of the user while typing the answer

themselves. The requirement that, the response to the challenge must be sent inside the video, serves to plug this loophole.

A practical challenge response needs to balance both security and usability aspects - that is, it should not be possible to solve it by using pre-recorded videos and it should require as less human cooperation as possible. Of course, the ideal case would be to have an entirely passive challenge response system where the response to the challenge can be automatically gauged without the user explicitly performing any action. Theoretically, a passive challenge can be presented for each video frame and the corresponding response can guarantee the freshness. In practice, this is difficult to implement and the only existing scheme we found for video required special lighting conditions to work.

As previous schemes were not suitable, we propose our own challenge response mechanism. The scheme requires the user to wave a piece of square paper in a particular part of the video frame. The actual location and the colour of the paper to be waved is randomised so that a replay attack can be thwarted. As this requires user cooperation, the challenge response should be used intermittently - such as when the video tampering detector reports a possible tampering. The reason behind this is that the initial frames are live and their freshness has been proved. For an attacker to perform a replay attack, an older frame from the same or other stream would need to be sent to the server. This would in turn result in a frame insertion forgery and would be detected by the video tampering detector. As such, it is sufficient to run the challenge response scheme only when there is an indication of an inter frame forgery.

#### 4.2.5 Integrator

The overall system integrator utilises the inputs from all the components to make the decision. Consider a frame  $x_k$ , which is the k-th frame in the sequence. The system calculates the following for it

$$p(x_k) = f(p(x_{k-1}), f_{vr}(x_k), f_{pd}(x_k), f_{cr}(x_k), f_{td}(x_k))$$

Where  $p(x_k)$  is the probability that as of the k-th frame, the user is being



continuously authenticated

$f_{vr}$  calculates the probability that the user in the frame is the genuine user

$f_{pd}$  calculates the probability that the user is physically present in the frame

$f_{cr}$  is the probability that the freshness of the frame is assured, that is it is not a pre-recorded frame

$f_{td}$  is the probability that the frame itself has not been tampered with.

$f()$  is the function which integrates the above functions and decides if the system at that time can guarantee that the user is being continuously authenticated.

It should be noted that it is not necessary for all of the components to be active at the same time. It is only at the beginning of the session that all four of them must be active as it helps to ensure that the initial frames are fresh, non-tampered and contain the authentic user. Later, the verifier needs to be active and even the challenge response component may be activated only on certain conditions. Thus, the actual functions and how the integrator integrates these values, depends on the implementation itself. We have implemented some of these functions in our proof of concept implementation.

### 4.3 Proof-of-concept implementation

We have designed a proof-of-concept system which uses our model and implements the four basic components described above. As this is a proof-of-concept system, we have chosen to specifically focus on preventing a replay attack through implementing a user friendly challenge response system and detecting inter-frame forgery. Accordingly, we have not implemented measures for liveness detection (to prevent spoofing attacks using fake biometrics) or intra-frame forgery detection (to prevent attacks based on frame content modification).

We implemented the system in C# using Microsoft Visual Studio 2013. The application was developed as a Windows Form Application using the .net framework. For image processing and computer vision related tasks, we used the library Emgu CV[1] which is a C# wrapper for OpenCV. We describe the implementation details of each component below

### 4.3.1 Verifier

The verifier can be thought of as a generic biometric system. In the ideal case, the verifier should be able to verify the authentic user and reject others. While it is preferable that the verification system is passive, it is not strictly required. For our implementation, we used face recognition using Local Binary Patterns (LBP). The algorithm for LBP is already implemented and accordingly we used the functions in the EMGU CV library to code the face verifier. In addition to the above, we also required that that user input a password as another level of verification. This ensures that 2 different types of authentication is performed and helps to decreases the risk of an intruder breaking in.

The verifier requires the user to enter an id (text, keyboard), password (text, keyboard) and present the face (recorded live, camera). The output of the verifier is boolean: “true” for a genuine user and “false” for an impostor. A “true” output is only possible when the presented face matches the id in the database and the input password is correct.

Unlike the other components, we did not evaluate the verifier. This is because the verifier is essentially a generic biometric authentication system which has been studied at length. As our implementation is focused more on preserving freshness and preventing replay attacks, we felt that it would be sufficient to use an existing method for the purpose of demonstration.

### 4.3.2 Presence Detector

The presence detector is supposed to check for the continuous presence of the user. It is also supposed to check for the liveness of the biometric to prevent spoofing attacks. As liveness detection in biometrics has been previously explored, we have not implemented it in our proof of concept implementation. We have rather focused on detecting whether a human face is present in each frame. This combined with the other components can guarantee continuous authentication.

Our presence detector consists of 2 parts (see figure 4.2). The first involves tracking the face of the user. We implemented this using the face detection functions in the EMGU CV library. Our second involved calculating the average

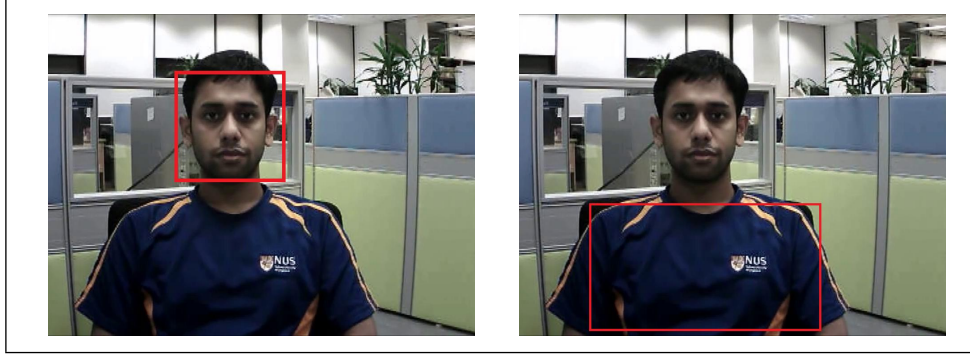


Figure 4.2: Screenshots of the presence detector showing the face tracking on the left and the t-shirt area on the right

colour of the upper part of the t-shirt of the user. For this, we selected a region below the face. The reason for using 2 of them is because sometimes the face detection algorithm may not be able to detect and track the face - possibly due the natural face movements by the participants.

In our implementation, we assumed that the user doesn't move from their chair. As such, we tracked the face on every frame and also noted the average colour of the t-shirt. We define a threshold of time  $t_1$  for which no presence (that is no face detected and no t-shirt colour match) is tolerated. We set this as 2 seconds (the value is arbitrary here). We similarly set another threshold  $t_2$  - the maximum time tolerable for which either of the conditions is satisfied. (This was set as 20 seconds, ten times the value of  $t_1$  here). Based on our observed data we use a score between 0 and 1. Initially the score is 1, meaning that the user is definitely present. As long as the score is  $> 0.5$ , we consider the user to be present. Accordingly,

1. Start with a score of 1
2. For every  $n$  seconds where face is tracked and colour of t-shirt matches,  
score = score
3. For every  $n$  seconds when face cannot be tracked AND colour of t-shirt doesn't match, score = score -  $(n*0.5/t_1)$   
(The value of  $n$  here depends on the frame rate)
4. For every  $n$  seconds where either face is tracked or colour of t-shirt matches  
score = score -  $(n*0.5/t_2)$

5. Once the score becomes 0.5 or less, the system automatically logs the user out.

The above implementation is a simple implementation. The thresholds calculated above were actually based on a limited number of observations but due to a lack of more data, they can be assumed to be arbitrary here. Unlike other systems where the score increases or decreases with input, our score linearly decreases and never increases. The reason for not increasing is because we anticipate that it is possible to fool the face detector as well as the t-shirt colour matcher. For example, in a scenario of a remote online examination, the initially authenticated user can move away from their chair and ask another user (wearing the same t-shirt) to take the exam on their behalf. As the face detector simply tracks the face and the colour matcher matches the rough colour, it may be possible for the users to exchange positions quickly. (This kind of user cooperation attack has been discussed later).

### 4.3.3 Challenge Response

In the previous section, we discussed that an ideal challenge response should be passive and require no human cooperation. In addition, the response to the challenge must be sent In band, that is, in the video itself.

As no suitable scheme was found, we designed our own challenge response scheme. The main idea behind this scheme is that the user will be asked to perform an action (from a set of actions). This action will be performed in front of the camera and the system should be able to decipher the response from the video frames recorded.

Our first experiment involved waving the hand. Given a random location on the frame (indicated by a box), the user would need to move their palm inside the box. The system would then calculate the average colour inside the box and check if it is close to the human skin colour. However, this proved to be harder than expected. We realised that skin colour differs from person to person and it would be hard to come up with an universal threshold (widening the threshold too much would render the system unable to differentiate between skin and non-skin). Another problem was that our background was beige coloured which was

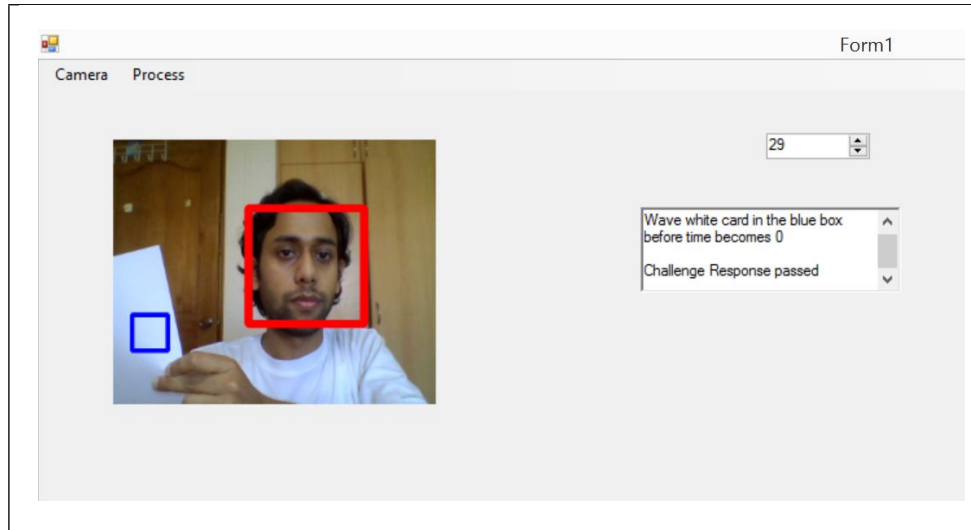


Figure 4.3: Screenshot of a successfully solved challenge response

automatically solving the challenge.

We realised that it would be helpful to expand the range of colours used. As such, we fixed 5 colours - white, red, blue, green and black - and accordingly places 5 differently coloured papers on the table. We conducted a pilot study involving 5 participants who were asked to solve the Challenge Response 10 times and asked to rate the scheme from 1 to 10 (1 as least user friendly and 10 was most user friendly). The average rating we received was 5.5. On being queried about the reasons, every participant mentioned that it would be easier to choose from 2 or 3 colours. Accordingly, we modified the scheme to only use 2 colours - black and white. Another feedback we received was that the location of the box sometimes obscured the face itself and moving the paper into the box was disruptive. Accordingly, we put limits on the box to appear on the edges of the frame instead of the centre where the user's face is presumed to be.

Our final implementation (see figure 4.3) works using the following steps:

1. Get a random coordinate in the frame (within the stated limits) and draw a blue bordered box. Display the box on the screen.
2. Get a random colour - black or white
3. Start a countdown timer to show time left to solve the challenge
4. For each subsequent frame

- (a) extract the region inside the box, convert to grayscale and calculate the average intensity. If the intensity is within the threshold for the colour, output true and end challenge, else output wait.
- (b) If time left reaches 0, output false.

Over here, true means that the challenge response has been successfully solved, wait means that it is in progress, while false means that the user has failed to respond.

#### 4.3.4 Video Tampering Detection

##### Approach 1

We implemented an existing work [6] as a base to start with. The aim of this work was to find instances of frame insertion and frame deletion based on the changes in optical flow. Their hypothesis was that any instance of frame deletion or insertion will cause an abrupt change in optical flow. Although the paper deals with both insertion and deletion, we only consider the deletion algorithm. This is because in the paper the authors have only considered insertion from a different video stream and mentioned that it is easier to detect than only deletion. In this paper the authors have suggested that in the event of a frame deletion, the optical flow between the two frames (before and after the deleted frames), is more than the average optical flow between continuous frames. The algorithm in brief is as follows:

1. Extract frames from the query video.
2. For each frame  $k$  and the next frame  $k+1$ , calculate the optical flow between these 2 frames.
3. Compare the value of optical flow between  $k$  and  $k+1$  with the average optical flow of its 4 adjacent neighbours (frames  $k-2, k-1, k+2, k+3$ ). If the value of optical flow between  $k$  and  $k+1$  is greater than  $t_1$  times the average of its 4 adjacent neighbours, do
  - (a) Calculate the average optical flow for all frames and check if the optical flow between  $k$  and  $k+1$  is more than  $t_2$  times the average. If it

holds true, then frame deletion has occurred.

Here  $t_1$  is the first threshold and for this implementation we have taken the value as 2, since it was suggested in the paper). Accordingly,  $t_2$  is 3.5, as suggested in the paper.

The above method essentially uses 2 thresholds which can be adaptive. We used C# along with the Emgu CV library to calculate the optical flow as described in above. This was then processed accordingly and a true/false result was obtained. True meant that the video was tampered while false meant it was not tampered. As we wanted to see the effect on genuine videos, we ran the algorithm on 100 instances of non-tampered videos and 100 tampered videos. The tampered videos were created by randomly deleting one-second worth of frames from the genuine videos. Our results indicated that 96 of the tampered were correctly detected as tampered while 32 of the genuine were also marked as tampered. This indicates a pretty high number of false positives.

## Approach 2

To reduce the false positives, we decided to use another mechanism of comparison mentioned in Bidokhti et. al [3]. Suppose  $OF(k)$  is the optical flow between frames  $k$  and  $k+1$  Then the ratio

$$\frac{2 \times OF(k)}{OF(k-1) + OF(K+1)}$$

should be usually close to 1 as the optical flow in a non-tampered video changes smoothly. We chose a maximum threshold of 3 for the above ratio based on observing values in genuine videos (the paper suggested a value of 2).

On evaluation of this, we observed that 93 of the tampered were correctly detected as tampered while 18 of the genuine were also marked as tampered. The number of false positives was still quite high but much better than the first approach. Accordingly, we decided to use this approach.

#### 4.3.5 Overall System Score

Overall System Score: The overall system runs frame by frame and maintains 2 scores, both of which are initialised to 1.

1.  $S_p$  : A score indicating presence
2.  $S_t$  : A score indicating tampering/freshness

For each frame the particular score is decreased accordingly.

1.  $S_p$  decreased according to the conditions in the presence detector
2.  $S_t$  decreased by 0.25, if the video tampering detection suspects a frame as not genuine.
3. If the value of  $S_t \leq 0.75$ , the challenge response module activates and requires an answer. If the challenge is successfully solved,  $S_t = S_t + 0.25$  and if not,  $S_t = S_t - 0.25$

We use a simple method to integrate the scores. At any given time, if either  $S_p \leq 0.5$  or  $S_t \leq 0.5$ , the system logs out the user.



## Chapter 5

# Evaluation and discussion

### 5.1 Experiments on Challenge Response

#### 5.1.1 Time to solve

We conducted an experiment to find out the time required by users to solve the challenge response.

Participants: For this experiment, we recruited the same 20 participants. All of our participants were university students between the ages of 19 to 26. Out of the 20 participants, 9 were female and 11 were male.

Setup: The participants were asked to sit at a table where a laptop with an inbuilt webcam was placed. White and black coloured paper was provided on the table. Each participant was asked to solve 10 challenges with a 30 second break between two consecutive challenges. The live video of the webcam recording of the participant was played (mirrored) on the screen. For the actual challenge, a box was overlaid on the video and the colour of the paper (black/white) was indicated on the screen. The participants were instructed to pick up the white or black paper as required and move it into the overlaid box to solve the challenge. The time interval between the start and successful solving of of the challenge was recorded.

We obtained a total count of 200 instances which we plotted as a histogram (see figure 5.1). Although the time was originally measured with the accuracy of one-thousandth of a second, we rounded each instance to a single decimal place for the purpose of binning. From the histogram we can see that in 194 instances

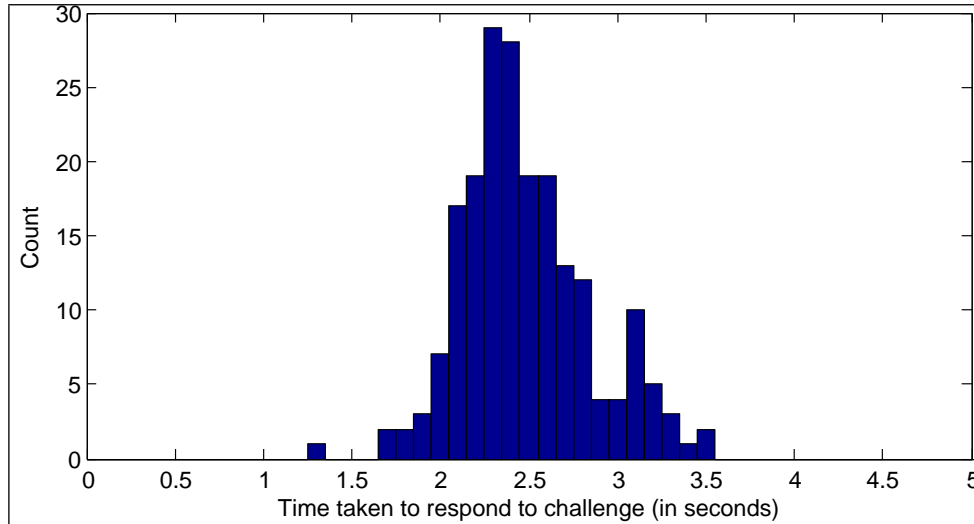


Figure 5.1: Histogram of time taken by participants to solve the challenge response

(97%), the time taken to successfully solve the challenge was 3.2 seconds or lower. The mean time to solve was 2.479 seconds while the median time was 2.4 seconds.

Based on the above results, we decided to accordingly keep 3.2 seconds as the maximum time allowed to solve the challenge. However, it should be noted that other factors could affect the time taken to solve the challenge. Our participants were university students in the age group of 19 - 28, who were instructed to solve the challenges quickly. It is possible that younger users may take more time to solve. In addition, the setup of the experiment (laptop, position on table etc.) may affect the time as well. As such, it is possible to customise this value accordingly.

We conducted a separate user study after fixing the time threshold as 3.2 seconds. The setup of this experiment was similar to the above: 20 participants were asked to solve the challenge response and each participant was asked to attempt it 10 times. The difference this time was that a countdown timer was displayed on the screen showing the amount of time left. The participants were seated on a chair with the background having a beige colour.

We noted that in 98.5% of instances, the participants were able to successfully solve it. In 3 cases however, they were not able to solve it within the time limit. We examined the reasons for this. In one case the participant simply

picked up the paper of the wrong colour which led to a delay and by the time the participant realised it, the timer had ended. However, as this was a single occurrence (we didn't observe this during the previous study of estimating the time required to solve), it is possible to consider this a one-off incident. The other two cases happened to lighting conditions. In both cases, the challenge required the participant to wave a black paper in the box. However, the paper was glazed and inadvertently reflected the overhead white light. This resulted in the software not considering it sufficiently close to a black colour and the time expired before the participants had the chance to correct the position of the paper and offset the effects of the reflection.

### **5.1.2 Discussion**

#### **Effect of lighting**

As we see above, the lighting may affect the result of the challenge response. This presents a possible usability problem in the long term. As a remote biometric system is supposed to be used by multiple users, it is impractical to assume that all users have the same lighting conditions or the same quality of paper. It is recommended that matte paper be used instead of glazed paper. Another solution to this would be to adjust the threshold for colour matching according to the user's situation. The user could be asked to wave a paper of a particular colour in the box and depending on the input received, the thresholds for colour matching could be adjusted. However, it should be noted that there should be some hard upper limits for adjustment (for example, the maximum 8-bit grayscale value for a black colour may not be allowed to exceed 100). This is to prevent gaming the system, where on being requested for a black paper, a malicious user might wave a white paper and the system could set the threshold for black at say, 256.

#### **Effect of background**

The background has a significant effect on the challenge response mechanism. As the challenges are generally presented close to the edges of the frame of view, a white background may be read as a correct response. This is clearly

problematic as it essentially compromises the challenge response system itself. We observed this problem when we initially tested the participants in front of a white wall and the colour of the wall was automatically solving the challenges. Our first option was to modify the mechanism to only present black challenges. However, this essentially reduced a degree of randomness and is not a good solution. One way to prevent this is to ensure that the user chooses a suitable background (outside the white and black colour matching thresholds). As it is easy to detect the background, the system can be programmed to refuse login if the background is not within a suitable colour range. However, this solution relies on the user and risks reducing the user friendliness and wide applicability of the system. Although we prefer the latter to the former (as it doesn't compromise the security), it is clear that a better solution to this needs to be found.

## 5.2 Experiments on Presence Detector

For evaluating the presence detector we did a simple evaluation. The 20 participants were asked to perform the following actions once each. The thresholds  $t_1$  and  $t_2$  were set as 2 seconds and 10 seconds respectively. We evaluated the time taken by the system to perform the necessary action in each scenario.

1. Leave the chair and walk out
2. Putting the head down on the table
3. Cover face with a paper magazine (containing a printed face)

### Case 1

Case 1: In all the 20 instances of case 1, the system correctly registered that the face was missing. In most cases, the system managed to logout in slightly more than the threshold  $t_1$  time. The average was 4.15 seconds.

In the ideal case, the system should have logged out the user within  $t_1$  seconds. However, due to the fact that most users took a couple of seconds to rise up and move away, the system continued to receive the input of the t-shirt colour.

## **Case 2**

Case 2: In all the 20 instances in case 2, the system correctly registered that the face was missing. The system actually managed to logout the user quickly in an average of 2.54 seconds.

We noticed that when participants put their head down on the table, it also obstructed their t-shirt. It is possible that when the system found both the t-shirt and the face missing, it quickly decreased the score and logged out.

## **Case 3**

Case 3: In all 20 instances, the system failed to log out and let the user continue. Of course, this is not desirable at all. However, the result is understandable, particularly because our implementation did not involve checking for liveness detection. As such, an attacker could simply wear a similar coloured t-shirt and cover their own face with a paper mask to access the system. Implementing liveness detection could possibly prevent this attack. Another way to mitigate this attack is to perform face recognition once in a while - for example maybe at regular intervals of 60 seconds. This would force the attacker to present the genuine user's face once in a while. In fact, instead of regular intervals, doing random checks might be more effective as it reduces the ability of the attacker to guess the time of attack.

### **5.2.1 Discussion**

As this was not our main focus, we did not do a detailed evaluation. From what we can see in the 3 cases above, it is clear that our proof of concept implementation has much room for improvement. Liveness detection is absolutely essential to prevent the type of attack in the third scenario.

## **5.3 Experiments on Video Tampering Detector**

### **5.3.1 Frame insertion and deletion**

For evaluating our tampering detection component, we recorded 5 videos each of the 20 participants while they were reading something on the screen. Each of

Table 5.1: Table showing accuracy of video tampering detector in frame insertion and deletion

	True positive	False positive
Frame insertion	45	9
Frame deletion	47	10

these video (total 100 videos) was 10 seconds long and did not contain any sudden motion by the participants. In about 50 of these videos we inserted frames from a previous part of the video while in 50 of them we deleted frames. The duration of inserted/deleted frame and point of insertion/deletion was random.

We notice that the algorithm is relatively good in spotting true positive cases (although it could be better) (see table 5.1). However, there is a lot to be desired in the case of false positives.

## 5.4 Frame freezing

In this attack, we randomly played the same frame multiple times (between 2 - 5) with the next frames continuing after that. This is problematic as it allows an attacker to delay the transmission. Unfortunately, our implemented algorithm was not able to detect a single instance. We then modified the algorithm to check for cases when the difference in optical optical flow is 0. Once this modification was done, the system was able to detect all 50 instances.

## 5.5 Alternative forward-reverse looping

In this attack we replayed frames but in a particular way. For example we played frames 1,2,3,4,5 and then 4,3,2,1 followed by 2,3,4,5 again. This technique preserves the actual optical flow differences in the genuine video and yet compromises the freshness. Our implementation was not able to detect this attack as there was no anomalous difference in optical flow.

## 5.6 Discussion

The results of using optical flow as a means of video tampering detection have not been as encouraging as previously expected. The high false positives are clearly

a limitation of this approach. A high false positive is problematic as it makes the system suspect a genuine user as an attacker and could potentially deny access to the user. More importantly, in our implementation, the challenge response gets activated in the case of suspected tampering. If this happens too frequently, as shown here, it can be disruptive to a user. This is of particular significance if the scenario is an online examination. An optical flow based method is desired because it solely depends on the content of the video. However, our experience suggests that one key problem is that there are no clear thresholds to accurately distinguish a genuine activity with a sudden change in optical flow from a case of tampering. As the thresholds of genuine and malicious activity overlap, it would be worth re-looking at the suitability of this approach.

## 5.7 Combined experiment

We performed an experiment to check the combined effect of our video tampering detection and challenge response. The steps of the experiment were as follows:

1. Assume the system has authenticated the user and is running. Switch the current stream and replace it with an older recorded video.
2. If the tampering detector manages to detect the attack, wait for the challenge response to engage
3. Once the challenge response starts, switch back to the live stream

The above experiment was to check if the video forensic and challenge response components were interacting as desired. We noticed the following

1. Out of 100 instances, the tampering detector detected 91 of them and the challenge response started. 9 instances were not detected.
2. All 91 attempted to switch the stream back to the live stream. 84 of them were detected by the tampering detector. (As this was the second detection, they were immediately logged out)
3. Out of the 7 left, 3 were able to correctly solve the challenge response and evade detection. 4 did not manage to solve it in time and were logged out.

If we do a total count, we realise that in 12 instances (9 initially and 3 later), the system was not aware that an intruder had taken control. Thus, it is very important for the tampering detector to have a sufficiently high true positive rate. In addition, we notice that in the 9 instances, the intruder continued to be logged in - as they provided a replay of the biometric information. A desirable feature to implement would be to periodically call the challenge response even if no threat has been detected. This ensures that even if an intruder manages to log in, after a certain amount of time, they would have to show their credentials again.



## Chapter 6

# Conclusion and future work

We have proposed a model for continuous biometric authentication which preserves freshness of the authentication information. Unlike previous approaches, our approach doesn't require the assumption that the sensors on the user's side are trusted. We have also proposed a new challenge response mechanism for video based authentication.

Of course, the implemented system is far from perfect. However, we did not aim to create a perfect system. Our proof-of-concept implementation is meant to demonstrate that it is possible to preserve freshness in continuous authentication systems. Practically trying to implement the model also helped us to understand the limitation in terms of usability. We believe this implementation is one step towards a more secure and usable system.

### 6.1 Future Work

Although the actual scope of this work is quite wide, we have identified some specific aspects which we feel can be worked upon in the near future

- **Intra-frame forgery detector** This is an important component which is missing in the current implementation. As there has already been quite a lot of research on image forensics, it would be worthwhile to try implementing an existing algorithm and then test the entire system.
- **Liveness detection** This is necessary for preventing spoofing attacks. Liveness detection techniques have been well studied in literature and it is

feasible to integrate into our implementation.

- **Decay factor** A time based decay factor (for the score) is useful to ensure that intruders who slip through the tampering detection algorithms, do not have indefinite access to the system. This concept is analogous to a session expiry time and has been previously explored [49] in the context of continuous authentication. Instead of logging out the user, the system may initiate a challenge-response at regular intervals or perform passive liveness detection.
- **Usability** In this work, we did not really focus on usability of the system as a whole. However, usability is critical to determine whether a system is practically deployable. Although there has been some work on usability[26], there is scope for more.

## 6.2 Open problems

- **Passive challenge response** As mentioned previously, making the challenge response operate independently of the user's cooperation would be a good advancement. It would enable the system to pose multiple challenges and mitigate the effects of the error rates of the tampering detection algorithms.
- **Defending against user cooperation attacks** One possible attack scenario happens where the authentic user cooperates with a malicious attacker. For example, in the case of an online examination, the user might want a proxy to appear. In such a case, the user might be physically present in front of the camera, but the screen is duplicated and shared with the attacker. An interesting direction of research would be to see if it is possible to guarantee that the space around the user is secure itself. This is challenging as the user and attacker may not actually be present in the same physical location. One approach could be to use multiple camera feeds with one camera pointing to the keyboard. The correlation between the input keystrokes and the keystrokes visible could then be examined.

# Bibliography

- [1] Emgu cv. <http://www.emgu.com>.
- [2] A. Altinok and M. Turk. Temporal integration for continuous multimodal biometrics. In *Proceedings of the Workshop on Multimodal User Authentication*, 2003.
- [3] A. Bidokhti and S. Ghaemmaghani. Detection of regional copy/move forgery in mpeg videos using optical flow. In *Artificial Intelligence and Signal Processing (AISP), 2015 International Symposium on*, pages 13–17, March 2015.
- [4] W. E. Burr, D. F. Dodson, E. M. Newton, R. A. Perlner, W. T. Polk, S. Gupta, and E. A. Nabbus. Sp 800-63-1. electronic authentication guideline. Technical report, National Institute of Standards & Technology, 2011.
- [5] C. M. Carrillo. Continuous biometric authentication for authorized aircraft personnel: a proposed design. Technical report, Naval Postgraduate School, Monterey, California, 2003.
- [6] J. Chao, X. Jiang, and T. Sun. A novel video inter-frame forgery model detection scheme based on optical flow consistency. In *International Workshop on Digital Watermarking*, pages 267–281. Springer, 2012.
- [7] G. Chetty and M. Wagner. Multi-level liveness verification for face-voice biometric authentication. In *2006 Biometrics Symposium: Special Session on Research at the Biometric Consortium Conference*, pages 1–6. IEEE, 2006.

- [8] J. A. Clark and J. L. Jacob. A survey of authentication protocol literature: Version 1.0. Technical report, 1997.
- [9] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann. Touch me once and i know it's you!: implicit authentication based on touch screen patterns. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 987–996. ACM, 2012.
- [10] J. Elson, J. R. Douceur, J. Howell, and J. Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. In *ACM Conference on Computer and Communications Security*, volume 7, pages 366–374, 2007.
- [11] E. Flior and K. Kowalski. Continuous biometric user authentication in online examinations. In *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*, pages 488–492. IEEE, 2010.
- [12] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *Information Forensics and Security, IEEE Transactions on*, 8(1):136–148, 2013.
- [13] S. Furnell, J. P. Morrissey, P. W. Sanders, and C. T. Stockel. Applications of keystroke analysis for improved login security and continuous user authentication. In *Information systems security*, pages 283–294, 1996.
- [14] A. Gironi, M. Fontani, T. Bianchi, A. Piva, and M. Barni. A video forensic technique for detecting frame deletion and insertion. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6226–6230, 2014.
- [15] R. Gossweiler, M. Kamvar, and S. Baluja. What's up captcha?: a captcha based on image orientation. In *Proceedings of the 18th international conference on World wide web*, pages 841–850. ACM, 2009.
- [16] G. Goswami, R. Singh, M. Vatsa, B. Powell, and A. Noore. Face recognition captcha. In *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pages 412–417. IEEE, 2012.

- [17] J. Holman, J. Lazar, J. H. Feng, and J. D'Arcy. Developing usable captchas for blind users. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, pages 245–246. ACM, 2007.
- [18] A. K. Jain, S. C. Dass, and K. Nandakumar. Can soft biometric traits assist user recognition? *Defense and Security*, pages 561–572, 2004.
- [19] A. K. Jain, A. Ross, and S. Pankanti. Biometrics: a tool for information security. *Information Forensics and Security, IEEE Transactions on*, 1(2):125–143, 2006.
- [20] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):4–20, 2004.
- [21] R. Janakiraman, S. Kumar, S. Zhang, and T. Sim. Using continuous face verification to improve desktop security. In *Application of Computer Vision, 2005. Seventh IEEE Workshops on*, volume 1, pages 501–507. IEEE, 2005.
- [22] A. T. B. Jin, D. N. C. Ling, and A. Goh. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern recognition*, 37(11):2245–2255, 2004.
- [23] Z. Jorgensen and T. Yu. On mouse dynamics as a behavioral biometric for authentication. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pages 476–482. ACM, 2011.
- [24] A. J. Klosterman and G. R. Ganger. Secure continuous biometric-enhanced authentication. Technical report, Carnegie Mellon University, 2000.
- [25] K. A. Kluever and R. Zanibbi. Balancing usability and security in a video captcha. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, page 14. ACM, 2009.
- [26] G. Kwang, R. H. Yap, T. Sim, and R. Ramnath. An usability study of continuous biometrics authentication. *Advances in Biometrics*, pages 828–837, 2009.

- [27] K.-Y. Lam and D. Gollmann. Freshness assurance of authentication protocols. In *Computer Security ESORICS 92*, volume 648 of *Lecture Notes in Computer Science*, pages 261–271. 1992.
- [28] B. Miller. Vital signs of identity [biometrics]. *IEEE Spectrum*, 31(2):22–30, 1994.
- [29] F. Monrose and A. D. Rubin. Keystroke dynamics as a biometric for authentication. *Future Generation computer systems*, 16(4):351–359, 2000.
- [30] K. Niinuma and A. K. Jain. Continuous user authentication using temporal information. *SPIE Defense, Security, and Sensing*, pages 76670L–76670L, 2010.
- [31] K. Niinuma, U. Park, and A. K. Jain. Soft biometric traits for continuous user authentication. *Information Forensics and Security, IEEE Transactions on*, 5(4):771–780, 2010.
- [32] F. P. NIST. Minimum security requirements for federal information and information systems. Technical report, National Institute of Standards & Technology.
- [33] N. K. Ratha, J. H. Connell, and R. M. Bolle. An analysis of minutiae matching strength. In *Audio-and Video-Based Biometric Person Authentication*, pages 223–228. Springer, 2001.
- [34] J. Roth, X. Liu, A. Ross, and D. Metaxas. Biometric authentication via keystroke sound. In *Biometrics (ICB), 2013 International Conference on*, pages 1–8. IEEE, 2013.
- [35] B. Schneier. Inside risks: The uses and abuses of biometrics. *Commun. ACM*, 42(8):136–, Aug. 1999.
- [36] M. P. Segundo, S. Sarkar, D. Goldgof, L. Silva, and O. Bellon. Continuous 3d face authentication using rgb-d cameras. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 64–69. IEEE, 2013.

- [37] T. Shanableh. Detection of frame deletion for digital video forensics. *Digital Investigation*, 10(4):350 – 360, 2013.
- [38] S. Shepherd. Continuous authentication by analysis of keyboard typing characteristics. In *Security and Detection, 1995., European Convention on*, pages 111–114. IET, 1995.
- [39] T. Sim, S. Zhang, R. Janakiraman, and S. Kumar. Continuous verification using multimodal biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):687–700, 2007.
- [40] D. F. Smith, A. Wiliem, and B. C. Lovell. Face recognition on consumer devices: Reflections on replay attacks. *IEEE Transactions on Information Forensics and Security*, 10(4):736–745, 2015.
- [41] M. Stanić. Continuous user verification using mouse dynamics. Technical report, Agency for Science and Higher Education,Zagreb, Croatia.
- [42] L. Sun, W. Huang, and M. Wu. Tir/vis correlation for liveness detection in face recognition. In *International Conference on Computer Analysis of Images and Patterns*, pages 114–121. Springer, 2011.
- [43] L. Von Ahn, M. Blum, N. J. Hopper, and J. Langford. Captcha: Using hard ai problems for security. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 294–311. Springer, 2003.
- [44] Q. Wang, Z. Li, Z. Zhang, and Q. Ma. Video inter-frame forgery identification based on consistency of correlation coefficients of gray values. *Journal of Computer and Communications*, 2(04):51, 2014.
- [45] Q. Wang, Z. Li, Z. Zhang, and Q. Ma. Video inter-frame forgery identification based on optical flow consistency. *Sensors & Transducers*, 166(3):229, 2014.
- [46] W. Wang and H. Farid. Exposing digital forgeries in video by detecting double mpeg compression. In *Proceedings of the 8th Workshop on Multimedia and Security*, pages 37–47, New York, NY, USA, 2006. ACM.

- [47] W. Wang and H. Farid. Exposing digital forgeries in interlaced and deinterlaced video. *IEEE Transactions on Information Forensics and Security*, 2(3):438–449, Sept 2007.
- [48] H. Xu, Y. Zhou, and M. R. Lyu. Towards continuous and passive authentication via touch biometrics: An experimental study on smartphones. In *Symposium On Usable Privacy and Security (SOUPS 2014)*. USENIX Association, 2014.
- [49] S. Zhang, R. Janakiraman, T. Sim, and S. Kumar. Continuous verification using multimodal biometrics. *Advances in Biometrics*, pages 562–570, 2005.