Biometric Video Game Authentication

Angel Ding, Gabriel Kammer, John Sragow, Sabina Tontici

ABSTRACT

As IoT (Internet of Things) has become more prevalent, behavioral biometric authentication schemes have been studied as an added layer of protection for devices. Unlike static authentication methods like passwords, PIN numbers or physiological bio-metrics like fingerprints, behavioral biometric authentication schemes use behaviors of users while interacting with devices. In this paper, we focus on analyzing the security of such authentication schemes as an added layer for VR headset authentication.

In Virtual Reality games, taking off one's headset to type in or to give a fingerprint to authenticate could cause visual damage to the end-user. Mustafa et al. (2018) proposed a ML authentication scheme that trains a machine learning classifier using player's body movements.

This authentication scheme is extremely useful for VR video games because it is both difficult and potentially hazardous for a user to remove their headset during gameplay. We suggest that this technology could also be used for console games with motion sensors like Nintendo Wii, because such systems also make it quite difficult to type in passwords.

In this paper, we attempt to analyze the security of this ML authentication approach by proposing a definition of CMA security, and analyzing if this scheme is CMA secure under our definition. We also propose extensions to existing schemes by adding support for additional "friends" of the owner to use the device given permission from the owner.

Finally, we propose a variation of this authentication scheme for user to user communication, where this biometric authentication method is used as a challenge to authenticate different users to each other. We also present a cryptographic analysis to this scheme based on our definition of CMA security.

Keywords: Biometric Authentication, VR games, CMA Security

1 INTRODUCTION

Machine learning authentications have been implemented in various devices and applications as an added layer of authentication. There has been lots of literature on the effectiveness of these machine learning algorithms. Bello et al. (2020) reported that the applications of machine learning authentication schemes can improve the security of mobile phone touch screen devices. However, there are lots of room for improvements since most machine learning algorithms heavily depend on feature extraction methods. This indicates that before using this technique, separate work for extracting critical features is required. Other surveys and analysis have been done on behavioral biometrics in healthcare and behavioral biometrics on touch dynamic based user authentication schemes, as Pryor et al. (2021) did in their experiment.

Even though lots of studies have shown that these machine learning algorithms are effective, there has been very little literature focusing on the security of those machine learning authentication schemes. In the following sections, we will present a cryptographic evaluation of the security of the machine learning authentication scheme for VR headsets.

Firstly, we must define the **headset**. In this context, the headset is a virtual reality device used to play games. While wearing the headset, motions are restricted to what the headset can sense you doing, which is arm motion, hand motion, and head motion. As such, typing or having to input a lengthy code can be difficult while wearing a headset. Because we do not want the user to have to take off their headset, we want our authentication scheme to be both **secure** and **convenient**.

Secure: We want to ensure that an unauthorized user cannot impersonate the owner of the headset without the owner's permission.

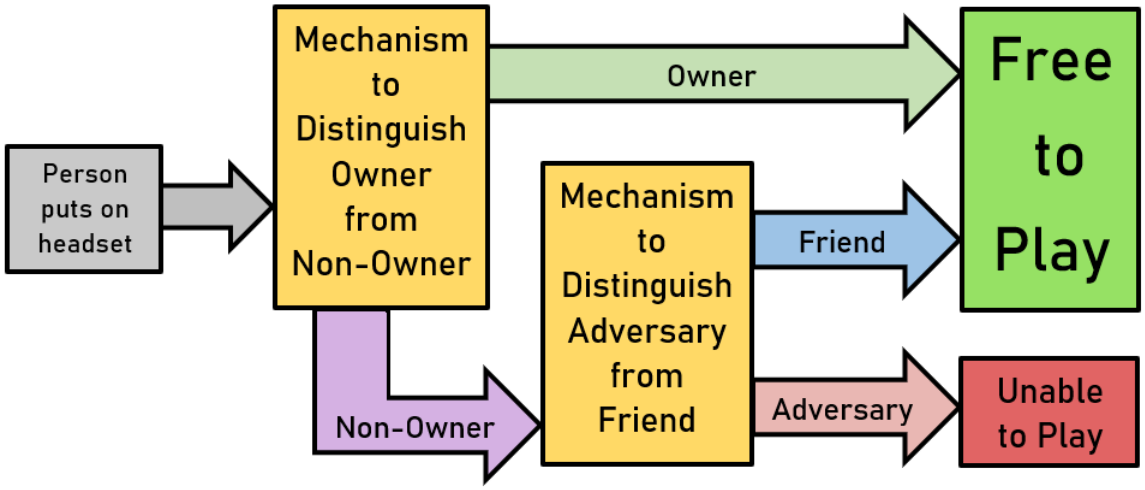
Convenient: We want to minimize how much an authorized user has to remove their headset in order to play.

In order to more accurately describe our model, we will define the actors involved:

- **Owner**: The owner of the headset, and the one who can always provide access to the headset. They should always be able to use the headset without having to remove it.
- **Friend**: A friend of the owner, who should not be able to use the headset without the owner's permission, but who the owner should, at any point, be able to allow to use the headset.
- **Adversary**: A person who wants to use the headset whom the owner does not want to be able to use the headset.
- **Non-owner**: A friend or an adversary

2 THREAT MODEL

Now that we have defined terms, we can outline our threat model.



The above chart describes how this system is supposed to work. Whenever anyone puts on the headset, it will immediately try to distinguish between the owner and anyone else. The owner will be free to play, while anyone else will have to go through another mechanism to distinguish a friend from an adversary. If that mechanism rules that the person wearing the headset is a friend, they are free to play. If it determines that they are an adversary, they are unable to play. This means we have two mechanisms to define:

1. Distinguish the owner from a non-owner.
2. Distinguish a friend from an adversary.

Thus, the two threats are defined as being able to crack the two mechanisms: a non-owner impersonating an owner, and an adversary impersonating a friend.

In practice, however, whether someone is an adversary or a friend is entirely up to the prerogative of the owner. Because this mechanism differentiating a friend from an adversary necessarily only happens when the owner is not wearing the headset, it is thus necessary and not inconvenient to ask the owner whether the person wearing the headset is a friend or an adversary, perhaps via a push notification to the owner's phone or computer containing a photograph of the face of the person wearing the headset. As such, this second mechanism is not an especially difficult problem to solve, as long as our first mechanism is successful. But this still begs the question: how can we distinguish between an owner and a non-owner?

With this as our question of focus, we can move on to defining what a secure scheme is.

CMA Security: We define a scheme as CMA secure if a non-owner can win the following game with negligible probability:

- What the non-owner can do:

  - See the owner authenticating themselves as many times as they want
  - Attempt to authenticate themselves as many times as they want

- The non-owner wins if they can successfully authenticate themself to play on the headset.

One example of how this security scheme may come into play is with a younger sibling. If you are the owner of a headset, your jealous little sibling might want to play for themselves, but you don't want them to, because you know they'll ruin all of your progress. This little sibling can watch you authenticate yourself over and over again, and break into your room as often as they want whenever you're not home to try and authenticate themself to use your headset. To make sure that sneaky kid can never commit heinous actions to your save file, you need to make sure that even given all of the information that they have about you and how you authenticate yourself to your headset, that they can never play the game unless you want them to.

Perhaps a more serious example of an application of this security scheme is to protect virtual property. In the past few years, we have seen the rise of virtual worlds and virtual property, and it is important that it not be possible for digital thieves or griefers steal one's virtual valuables.

3 THE LOCAL SCHEME

Our scheme will be based on using biometrics to authenticate the owner. We plan to train a logistic regression classifier based on the movements of the user's hands, arms, head, and eye using different sensors of the headset. When the user buys a new VR headset and use it for the first time, the user is required to put in a password to register. The headset will then collect their body movements as behavioral patterns as the user traverses in the virtual reality space. We will use feature extraction to train the classifier for the user to authenticate later without taking off their headsets. In the following paragraphs, we will describe the process, including and how the owner and the friend would authenticate:

- When a user starts playing a game, the headset will monitor their behavioral patterns and collect data from their body movements. If that data continually passes the threshold of the linear classifier, we conclude that the current user is the owner and allow them full access. Otherwise, if the headset detects the user's body movements continually fail to pass the threshold of the linear classifier, the headset will prompt the user to pass a second authentication scheme whereby a push notification will be sent to another device owned by the owner.
- If the owner decides to allow the user to play, and the new user will have their continuous authentication validated based on a fresh classifier.
- Otherwise, the device will be locked.
- When a non-owner authorized user (friend) is playing the game, a new classifier will be trained and continuously authenticate the new authorized user.
- **Continuous Authentication**: Throughout the whole time when a user is playing the game, the headset will continuously monitor the user's behavioral patterns and collect data from their body movements. It will then go through the same steps to determine if a user should be allowed to use the headset.

4 LOCAL SCHEME ANALYSIS

There are several potential attacks against this scheme which attempt to impersonate the owner of the headset. For each of these cases, we consider Alice to be the owner of the headset, and Eve to be an adversary whom Alice does not want using the headset.

Lunchtime Attack on Owner: Alice has already authenticated her headset, and she decides to take off her headset to go eat lunch. However, she has forgotten to lock her console! Eve comes along and puts on the headset to play it for herself. How does this scheme prevent such an attack?

- In this attack, Eve simply puts on Alice's authenticated headset.
- Thanks to continuous authentication, the device will detect that Eve's biometric patterns do not match Alice's, and lock Eve out of the device. At the same time, Alice will get a push notification saying that someone, in this case Eve, is trying to use her headset.

Lunchtime Attack on Friend: Alice has already authenticated her headset, and she decides to let her friend Bob play. Bob puts on the headset, and is soon thereafter confronted with a screen telling him that he must wait for Alice's approval to continue playing the game. Alice gives her approval, and Bob is free to play the game. After a while of playing, Alice and Bob leave to go get lunch, but they have once again forgotten to lock the headset. Eve comes by once again to play on the headset for herself. How does the scheme prevent such an attack?

- In this attack, Eve simply puts on a device that Alice authenticated for Bob to use.
- Despite the fact that Bob is not the owner, for the period while Bob was playing, the device created a new classifier for Bob's movements.
- This classifier will detect that Eve's movements are not Bob's, and will lock Eve out of the system. At the same time, Alice will also get a push notification that someone else tries to use her headset.

Computer-Based Impersonation: Eve now has a very powerful robot which can analyze video footage of Alice playing with her headset. Using that information, the robot can analyze what types of movements will pass the linear classifier. The robot can thus be inserted into the headset and perfectly capture Alice's body movements, impersonating her. How does this scheme prevent such an attack?

- The headset completely covers the eyes of the user, so no onlooker or video of the user from outside the headset can see the movements of the user's eyes while they are wearing the headset.
- If we assume that communications between the headset's motion sensors and the device performing the authentication are secure, it is not possible for the robot to have any information about how Alice's eyes move during gameplay.
- It is thus impossible for the robot to create a scheme whereby eye motions are accounted for, and it is thus impossible for the robot to impersonate Alice.

We have thus proven that none of the above attacks will work against this scheme. But how can we prove that there isn't some other attack which we have yet to come up with which can break this security scheme?

Unfortunately, we cannot do that with our current definition of CMA security, due to attacks such as the following:

Window Smash Attack: To describe how this attack works, we first need to describe the inner workings of the continuous authentication scheme in the Mustafa paper. The features of the movement recorded by the motion sensors are broken up temporally into small sections called **windows**. Each individual window is what is tested by the classifier: The continuous authentication works by continually running . The classifier uses a threshold j, such that:

- Each window is run through the logistic regression classifier as it is generated by the user's motions.
- The classifier outputs a number which we can treat as a probability that this window was generated by a valid user. If this probability is greater than the threshold j, then it is accepted as a correct window.
- Otherwise, it is labelled as an incorrect window.
- If the classifier finds that the majority of the previous k windows were labelled as incorrect by the classifier, then it will lock the headset.

For this attack, choose a random feature vector as a window w to be plugged into the classifier. By how we defined j, if the classifier determines the probability of w being created by the user to be greater than j, then w is accepted as correct. There is thus a $1-j$ chance that a randomly chosen window is accepted as correct. Thus, if all of the adversary's eye movement windows are w, then there is a $1-j$ chance that the adversary will be authenticated. Because in most cases, eye movement does not affect movement in virtual space, and all of the other motion factors in virtual space can be impersonated with a Computer-Based Impersonation attack, the Window Smash attack gives the Computer-Based Impersonation attack a $1-j$ chance of successfully cracking the authentication, and allowing the Computer-Based Impersonation to do whatever it wants.

But how do we know that $1-j$ is not negligible? If $1-j$ were negligible, that would mean that j is extremely close to 1. That would make it extremely unlikely for legitimate windows produced by the owner to be marked as correct, which would mean that the owner is very likely to be constantly locked out of their own headset, interrupting their gameplay and forcing them to remove the headset repeatedly to authenticate themself. In other words, the headset would be practically unusable if j were too close to 1. This means that $1-j$ cannot be negligibly small, which means the Window Smash attack can be used to authenticate an adversary with non-negligible probability. Thus, this scheme is *not* CMA secure, according to our stated

definition of CMA security.

However, our definition of CMA security is quite broad. Although the Window Smash attack demonstrates a major flaw in this local security scheme, there may still be methods to counteract it that fall outside of our definition of CMA security:

- Despite the fact that $1 - j$ cannot be negligible to the point of computational insignifiance, it can still be very small. This means that if we make it very difficult for an adversary to attempt to authenticate themself a large number of times, perhaps through a device that which completely disables the device after too many consecutive failed authentication attempts (like what iPhones have), then it can be nearly impossible for this attack to succeed. However, this contradicts the CMA security assurance that the adversary can try to authenticate themself as much as they want.
- A system which detects seemingly robotic movements that completely locks up the headset until the owner allows it to run again is one such potential solution, but this too contradicts the CMA security assurance that the adversary can try to authenticate themself as much as they want.

Unfortunately, though this scheme can withstand a multitude of attacks, it cannot be CMA secure.

5 THE USER-TO-USER SCHEME

In this variant of the scheme, we use a biometric challenge to authenticate a message sent from Alice to Bob. Suppose that Alice and Bob would like to communicate in a virtual reality setting securely. We propose the following authentication scheme:

- On the first encounter between Alice and Bob, they use a standard cryptographic protocol to exchange secret keys, and can from then on communicate with standard guarantees on encryption and authentication. Additionally, Alice and Bob send each other the classifiers for their movement patterns, which Alice and Bob store. on their devices.
- When Alice and Bob want to communicate with one another, they can use standard cryptographic protocols to ensure that they are talking to someone with access to all of their shared secret keys. However, Alice does not know whether she is talking to Bob, or someone else using Bob's headset. To verify this:

    1. Alice chooses n points in virtual space, and sends those points to Bob.
    2. Bob then moves to each of those points sequentially and sends his biometric data from that movement to Alice.
    3. Alice verifies that Bob's classifier from their first encounter identifies that this biometric data was generated from Bob, and that the movements from one point to another were within a small degree of tolerance of a straight line (to prevent a random-walk type scenario).

6 USER-TO-USER SCHEME ANALYSIS

We first make a few assumptions about what an adversary cannot do with respect to the machine learning classifier:

- An adversary cannot generate artificial biometric data that matches another person's classifier.
- A classifier can catch a "spliced" segment of biometric data composed of multiple separate segments of biometric data from someone.
- A classifier detects biometric data from exactly one person.

Now we will show that this scheme is secure against an impersonating attacker. More formally, for any adversary Eve distinct from Bob, the probability that Eve successfully authenticates as Bob is negligible on security parameter n, even if Eve is able to observe Alice and Bob performing authentication.

Suppose that there are m options for points in virtual space for Alice to choose. Then if Alice chooses a sequence of n such points, there are m^n total paths that Alice could pick. Alice chooses one of them and sends it to the second party (either Bob or Eve).

If the second party is Bob, then Bob can simply move to each of the points in order and send the biometric data of him doing so to Alice. Alice checks that this is indeed Bob's biometric data and that the data lines up with him moving to all of the

required points. Alice then accepts Bob with probability 1.

If the second party is Eve impersonating Bob, then Eve must attempt to send a log of biometric data that matches reaching each of the sequence of points. Eve is assumed not to be able to generate artificial biometric data nor splice together pieces of Bob's biometric data. Therefore Eve's only option is to send a previously collected sequence of biometric data collected from Bob that goes through each of the points that Alice sent. However, since this sequence of randomized in each authentication attempt, the probability that a previous authentication attempt contained exactly a sequence of movement that went through each of the existing points in straight lines is approximately $\frac{1}{m^n}$. This is negligible with respect to n, so the probability that Eve succeeds is negligible. Hence, our scheme is secure against an impersonation attack.

7 FUTURE WORK

In the following paragraphs, we introduce some potential research directions for future experiments and some applications of this machine learning scheme for games outside virtual reality.

Consider a large, multiplayer online VR game (MMOG) in which many players move around in virtual space. Let Alice be a player of this game. However, perhaps due to some poor security choices by Alice, Eve has managed to take complete control of Alice's VR headset, bypassing any local authentication, and obtaining all of Alice's passwords and secret keys related to the VR headset. Is there any way for Alice to get her headset back?

Consider that Alice and Bob would like to communicate securely over virtual reality. Traditional cryptography can ensure that Alice is talking to someone using Bob's VR headset. However, Alice does not have a way to verify that someone else, say Eve, has not taken physical control of Bob's headset in some way. How can we use biometric authentication to ensure that Alice is talking to Bob instead of a physical impostor?

We also present the possibility of using this scheme for console games with motion sensors like Nintendo Wii. It's not user friendly to ask end-users to control the cursor on the screen with a controller. Therefore, a machine learning authentication based on behavioral patterns could be helpful.

8 CONCLUSIONS

In previous sections, we defined a local machine learning authentication scheme and proposed a definition for CMA security for it. Our scheme trains a logistical regression model based on the owner's eye movements, body movements and head movements. To distinguish an owner from non-owners, it does continuous authentication while the owner or an authorized user is playing VR games. To distinguish a friend from an adversary, we will send the owner a push notification if the system thinks a non-owner is trying to authenticate. However, through our analysis, we concluded that even though our scheme is resistant to a wide range of attacks, it's still not CMA secure. Then, we extended the application of this machine learning authentication scheme to a user-to-user communication scheme. In this communication scheme, the machine learning authentication is sent from Alice to Bob as a challenge. In this case, we concluded that the user-to-user communication scheme is CMA secure according to our definition of CMA security. At the end, we suggested further investigations of some security problems of this scheme and potential applications in console games.

REFERENCES

Bello, A. A., Chiroma, H., Gital, A. Y., Gabralla, L. A., Abdulhamid, S. M., and Shuib, L. (2020). Machine learning algorithms for improving security on touch screen devices: a survey, challenges and new perspectives. *Neural Computing and Applications*, 32(17):13651–13678.

Mustafa, T., Matovu, R., Serwadda, A., and Muirhead, N. (2018). Unsure how to authenticate on your vr headset? come on, use your head! In *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics*, pages 23–30.

Pryor, L., Dave, D., Seliya, D., Boone, D., Sowells, E. R., et al. (2021). Machine learning algorithms in user authentication schemes. *arXiv preprint arXiv:2110.07826*.