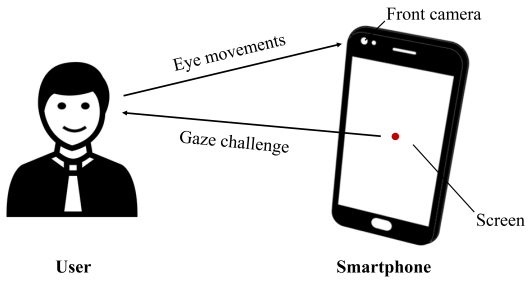


Fig. 1. System model of EyeAuth.

I. Introduction

Biometric authentication methods have become widespread on mobile devices. However, the two widely deployed schemes, i.e., face and fingerprint authentication, rely on users' sensitive physiological traits, raising serious privacy concerns [1], [2]. To alleviate the concerns, eye movement authentication has been proposed as a promising approach [2], [3] for the following reasons: i) eye movement exhibits traits distinctive enough that can reliably discern among a large group of individuals [4]; ii) eye movement rarely raises users' privacy concerns [2], while face and fingerprint are widely employed for law enforcement in many countries.

Simple eye movement authentication can be implemented with an eye tracker [4], [5], [6], [7]. But the additional device is expensive and unavailable on mobile devices. Some existing schemes utilize the mobile device's built-in camera for eye movement authentication. For example, EyeVeri [8] identifies users by capturing their gaze patterns while observing images using the smartphone camera. However, EyeVeri is vulnerable to replay attacks due to its reliance on static gaze challenges. Moreover, it suffers long-term performance degradation as a user's gaze behavior changes with familiarity to these images [4].

In this work, we propose EyeAuth, a replay-resistant authentication system based on reflexive eye movements. During authentication, the proposed system creates a random One-Time Gaze Challenge (OTGC) on the smartphone screen and captures the user's reflective eye movements with the front camera, as shown in Figure 1. EyeAuth is based on the fact that the reflexive reactions, considered as effortless neuronal responses [9], are less dependent on an individual's momentary conscious states [4], thus suitable for long-term authentication. Moreover, by comparing the OTGC pattern with the user's gaze trajectory, EyeAuth can verify if the eye movement is from a live user, thereby preventing replay attacks.

Although conceptually intuitive, EyeAuth faces two design challenges. Firstly, existing gaze estimation techniques on mobile devices yet have insufficient precision (i.e., the prediction error is around 1.57cm [10]). The error can disrupt the captured gaze trajectory and further affect the liveness verification. Secondly, the random gaze challenge leads to varying eye movement responses for each authentication, making it hard to extract stable features.

To tackle the first challenge, we partition the screen into multiple regions (i.e., 12 in our later experiments) that are large enough to tolerate the gaze estimation error. Then we design the OTGC comprising a short-lasting dot that randomly appears multiple times in each region, which can trigger noticeable reflexive eye movements and ensure a clear gaze trajectory. For the second challenge, we find that although the overall gaze trajectory is random, its sub-trajectories (i.e., the eye's acceleration and deceleration process with the dot appearing) exhibit certain stability. We identify hand-engineered features from these sub-trajectories and develop a Self-Attention-based Siamese Network (SASN) with a specific training sampling strategy, enabling the extraction of dot position-irrelevant features. Our contributions can be summarized as follows.

- We propose EyeAuth, a smartphone authentication system based on reflexive eye movements. We design OTGC, a unique gaze challenge that ensures a clear gaze trajectory under gaze estimation errors, enabling replay attack resistance.
- We design SASN, a model that extracts dot position-irrelevant features based on self-attention and Siamese network. Incorporating hand-engineered features, EyeAuth can achieve reliable user authentication under random gaze challenges.
- We conduct extensive experiments with two smartphones and verify EyeAuth's effectiveness under different scenarios. The results demonstrate that it achieves a high balanced accuracy of 95.56% while preventing 97.39% of replay attacks.

II. Preliminary

A. Threat Model

An adversary's goal is to cheat EyeAuth to bypass the authentication. We consider three different types of attacks according to the specific professional knowledge and technical capabilities that an adversary could possess.

- *Blind attack.* An adversary is not familiar with the authentication mechanism of EyeAuth. After the gaze challenge appears, the adversary randomly observes the smartphone screen.
- *Impersonation attack.* An adversary is capable of observing both the victim's eye movements as well as the screen gaze challenge pattern when he/she is passing the authentication. The adversary tries to follow the gaze challenge with his own eyes as the victim does to pass the authentication.
- *Replay attack.* The adversary acquires the victim's authentication video and replays it to the victim's device. We consider two types of replay attackers: 1) the *video display attacker* plays the authentication video in front of the target device camera with a screen; and 2) the *video injection attacker* directly injects the authentication video into the system's data capturer. This attack is expected to achieve maximum replay attack performance as it avoids additional distortion that may occur during camera recording.

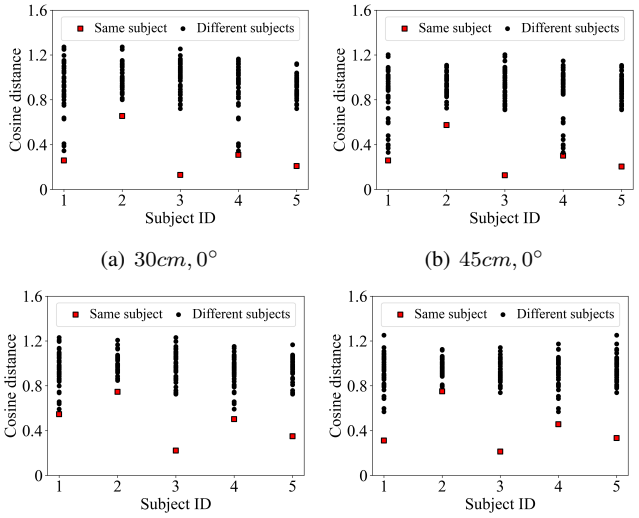


Fig. 2. Cosine distances of samples' gaze angular speed under the same and different subjects.

B. Feasibility Study

We conduct a feasibility study to investigate the uniqueness of human reflexive eye movements. Specifically, we ask five subjects to observe an identical gaze challenge displayed on a smartphone (Samsung Galaxy S10), which involves a dot changing positions in a predetermined order. Four scenarios, with different shooting distances and camera angles, are considered: 30cm, 0°; 45cm, 0°; 30cm, -10°; and 30cm, 10°. For each scenario, 10 samples are collected from each subject. We calculate the cosine distances of samples' gaze angular speed for the same and different subjects under each scenario, as shown in Figure 2. In each subject's column, a red square represents the average distance among the subject's own samples, while black dots represent the distances to samples from other subjects. Overall, we observe that samples from the same subject show smaller distances, regardless of different shooting distances or camera angles. This demonstrates that reflexive eye movements are consistent within the same subject and vary significantly between subjects, enabling the possibility of user authentication.

C. Design Goals

To meet system usability and security, we consider the following design goals.

- *Low cognitive load.* The system should not put too much cognitive load on the user. Ideally, users do not need to remember any credentials, carry tokens, or learn new procedures.
- *Universally applicable.* The system could be deployed on most smartphones without requiring installing new hardware components.
- *Resistance against replay.* The system should be difficult for an adversary to bypass by replaying biometric samples from a legitimate user.

III. System Design

In this section, we illustrate the design details of EyeAuth.

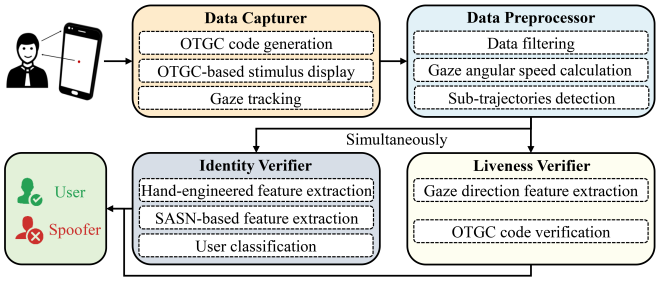


Fig. 3. System overview of EyeAuth.

A. Overview

EyeAuth operates in two phases: enrollment and authentication. During enrollment, the system creates a user profile using eye movement data captured by the front camera. During authentication, the profile serves as a template for matching the incoming user data.

As shown in Figure 3, EyeAuth consists of four modules: data capturer, data preprocessor, liveness verifier, and identity verifier. The data capturer monitors screen wake-up events in the background. Upon detecting an access request, the data capturer generates a one-time gaze challenge and displays it on the screen. It utilizes the front camera to record user eye movements at the same time and estimates gaze points on the screen plane. The gaze points are processed by data preprocessor for data filtering, gaze angular speed calculation, and sub-trajectories detection. Liveness verifier then extracts gaze direction features and verifies the OTGC code to determine whether the data is from a live user. Simultaneously, identity verifier extracts hand-engineered features and SASN-based features, which are further utilized for user classification. An individual who successfully passes both verifiers leads to access permission.

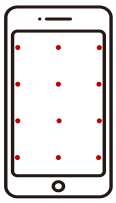
B. Data Capturer

We begin by introducing the gaze challenge designed and then illustrating the gaze tracking method employed.

1) OTGC Code Generation: To align with the design goals, we consider the following design criteria for the gaze challenge: i) it should be simple enough to alleviate the cognitive load on users; ii) it should be able to trigger noticeable eye movements that can tolerant gaze estimation error; iii) it should be unpredictable to prevent replay attacks.

To meet the first requirement, we adopt a simple dot that changes position multiple times as the gaze challenge. Specifically, when a dot appears on the background screen suddenly, the user's eyes will gaze towards it naturally. Such eye movements are fully reflexive [4], requiring no specific instructions or high cognitive effort from the user: his/her eyes do the work themselves.

To meet the second requirement, we preset twelve positions (as shown in Figure 4(a)), denoted by $C = \{c_1, \ldots, c_{12}\}$. By setting the coordinate of c_1 to $(0, 0)$, we derive the coordinate of other positions in Figure 4(b). As the screen of most mainstream commercial smartphones exceeds 5 inches [11], the chosen 12 fixed positions ensure the distance of adjacent



(a)

c_1 $(0,0)$	c_2 $(1,0)$	c_3 $(2,0)$
c_4 $(0,1)$	c_5 $(1,1)$	c_6 $(2,1)$
c_7 $(0,2)$	c_8 $(1,2)$	c_9 $(2,2)$
c_{10} $(0,3)$	c_{11} $(1,3)$	c_{12} $(2,3)$

(b)

Fig. 4. Visualization (a) and coordinates (b) of preset dot positions on the smartphone screen.

positions exceeds 1.57cm (i.e, the prediction error of state-of-the-art 2D gaze estimation techniques [10]). For example, the distance between adjacent positions is higher than 2.5cm even for a 4-inch phone with a screen size of 9.12cm×5.14cm.

To meet the third requirement, the gaze challenge is a one-time gaze challenge. The generated OTGC code could be represented as $P = \{p_1, \ldots, p_K\}, p_i \in C$, where K is the code length. Our design generates a huge pool containing $12 \times 11^{K-1}$ OTGC codes that guarantee each gaze challenge is unique.

2) Gaze Tracking: We employ a software-based gaze tracking algorithm [10] that runs in the background while presenting a gaze challenge. This algorithm demonstrates optimal performance in 2D gaze estimation (with a prediction error of 1.57cm) and remains stable at a long shooting distance (e.g., 180 cm). It takes a video $\{pic_1, \ldots, pic_n\}$ containing n images of human faces (captured by smartphone front camera with sampling rate f_s) as input and outputs two sequences: a gaze point sequence $POG = \{pog_1, \ldots, pog_n\}$ and a face-screen distance sequence $R = \{r_1, \ldots, r_n\}$. Here, pic_i is the i_{th} frame of the video. pog_i is a point-of-gaze, denoted by a vector $[x_i, y_i]^T$, where x_i and y_i correspond to the coordinates along the x and y axes on the screen, respectively. r_i is the distance between the center of the human face and the smartphone screen. Figure 5 illustrates an example of raw gaze points.

C. Data Preprocessor

We employ data filtering to enhance the gaze data quality and then calculate the gaze angular speed. Using the speed, we derive the sub-trajectories from the gaze trajectory.

1) Data Filtering: We use a Savitzky-Golay (S-G) filter to smooth the extracted gaze point sequence [12]. S-G filter is based on local polynomial least-square fitting in the time domain. We set the fitter window width as W, sliding it over $POG_x = \{x_1, \ldots, x_n\}$ and $POG_y = \{y_1, \ldots, y_n\}$, respectively. A second-order polynomial is employed to fit the data of POG_x and POG_y within the window. After filtering, we obtain $POG_{\overline{x}} = \{\overline{x}_1, \ldots, \overline{x}_n\}$ and $POG_{\overline{y}} = \{\overline{y}_1, \ldots, \overline{y}_n\}$. As shown in Figure 5, the circular points are the filtered points. We observe that the S-G filter reorganizes the noisy raw gaze points, revealing the user's gaze trajectory.

2) Gaze Angular Speed Calculation: The user's gaze angular speed can be calculated in two steps. We first calculate the gaze angle sequence $\Theta = \{\theta_1, \ldots, \theta_{n-1}\}$ from $POG_{\overline{x}}$, $POG_{\overline{y}}$, and R. Here, $\theta_i = 2arctan(d_i/2r_i)$, where d_i is the distance between $[\overline{x}_i, \overline{y}_i]^T$ and $[\overline{x}_{i+1}, \overline{y}_{i+1}]^T$, r_i is the corresponding face-screen distance. Note that this relationship holds only when the gaze angle is small [13] (e.g., observing a smartphone screen). Then, we calculate gaze angular speed sequence $\dot{\Theta} = \{\dot{\theta}_1, \ldots, \dot{\theta}_{n-2}\}$. Here, $\dot{\theta}_i = (\theta_{i+1} - \theta_i)f_s$. Algorithm 1 presents the step-by-step process of the gaze angular speed calculation.

Assuming the gaze point sequence has a length of n, we then analyze the time complexity of the algorithm. Specifically, the algorithm comprises two loops: one for calculating gaze angle and the other for gaze angular speed. The first loop runs $n-1$ times, and within the loop, the calculations for d_i and θ_i are constant-time operations, resulting in a time complexity of $O(n)$. Similarly, the second loop runs $n-2$ times, also featuring constant-time operations for calculating $\dot{\theta}_i$, leading to a time complexity of $O(n)$ as well. As a result, the total time complexity of Algorithm 1 is $O(n)$.

Algorithm 1 Gaze angular speed calculation.

Input: $POG_{\overline{x}}, POG_{\overline{y}}, R, f_s$
Output: $\dot{\Theta}$

for $\overline{x}_i, \overline{x}_{i+1} \in POG_{\overline{x}}, \overline{y}_i, \overline{y}_{i+1} \in POG_{\overline{y}}, r_i \in R$ **do**
  $d_i = \sqrt{(\overline{x}_{i+1} - \overline{x}_i)^2 + (\overline{y}_{i+1} - \overline{y}_i)^2}$
  $\theta_i = 2arctan\frac{d_i}{2r_i}$
  Append θ_i to Θ
end for
for $\theta_i, \theta_{i+1} \in \Theta$ **do**
  $\dot{\theta}_i = (\theta_{i+1} - \theta_i)f_s$
  Append $\dot{\theta}_i$ to $\dot{\Theta}$
end for

3) Sub-trajectories Detection: As the user's eyes track the random gaze challenge, the resulting gaze trajectory appears random. However, we observe that each large eye movement involves acceleration and deceleration two stages, revealing a certain degree of stability. Inspired by [14], we design a multi-threshold method to detect these sub-trajectories. Specifically, we first set a threshold T_h on $\dot{\Theta}$, deriving segments large than T_h and identifying their peaks. Subsequently, starting from each peak, we traverse along the slopes on both sides of $\dot{\Theta}$ to search for the start and end points of each sub-trajectory, which are defined as the first valley point smaller than T_l $(T_l < T_h)$. We discard segments with peaks higher than T_b $(T_b > T_h)$ or length shorter than L. The former may arise from blinking, while the latter could be induced by noise. Figure 6 illustrates the detection results. Note that most sub-trajectories are triggered by dot position change, while a small portion also arises from the user's inadvertent eye movements, such as glancing toward the center of the smartphone screen.

D. Liveness Verifier

Considering each OTGC code is uniquely defined by K dot positions, we verify the liveness by determining whether the user's eyes capture the dot timely in most positions.

1) Gaze Direction Feature Extraction: We extract gaze direction features to characterize the user's eye movement directions triggered by the gaze challenge. Specifically, these

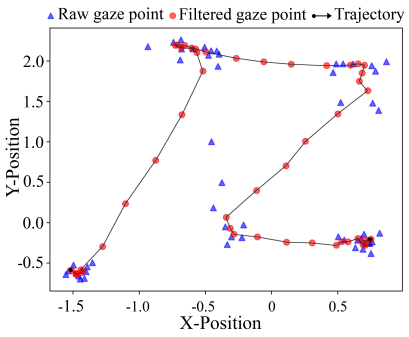


Fig. 5. Visualization of data filtering.

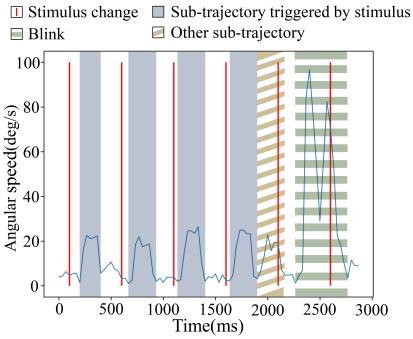


Fig. 6. Visualization of sub-trajectories detection.

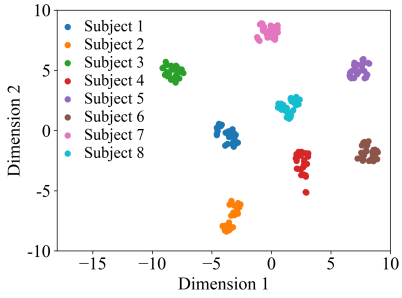


Fig. 7. t-SNE visualization of our features.

movements typically correspond to the sub-trajectories nearest to the moment of dot position change, as depicted in Figure 6. Each of these sub-trajectories consists of a sequence of gaze points, with its first and last points denoted by $[\overline{x}_i, \overline{y}_i]^T$ and $[\overline{x}_{i+j}, \overline{y}_{i+j}]^T$. The gaze direction feature could be represented as $\vec{u} = [\overline{x}_{i+j} - \overline{x}_i, \overline{y}_{i+j} - \overline{y}_i]^T$. For a OTGC code represented as $P = \{p_1, \ldots, p_K\}, p_i \in C$, we can extract K gaze direction features, denoted as $\{\vec{u}_1, \ldots, \vec{u}_K\}$.

2) OTGC Code Verification: We employ a challenge-response matching method to verify the OTGC code. Suppose that the dot consecutive appears in two positions $p_{n-1}(x_{n-1}, y_{n-1})$ and $p_n(x_n, y_n)$, we regard the dot position change $\vec{v}_n = [x_n - x_{n-1}, y_n - y_{n-1}]^T$ as a challenge. The gaze direction feature $\vec{u}_n$ associated with $\vec{v}_n$ is considered as the response. If the angle between $\vec{u}_n$ and $\vec{v}_n$ is less than α, we believe that the challenge and response match and the user's eyes have captured p_n. The process could be formalized as:

$$gazed(p_n) \Leftrightarrow \exists(\vec{u}_n, \vec{v}_n) : arccos\frac{\vec{u}_n \vec{v}_n}{|\vec{u}_n|\,|\vec{v}_n|} < \alpha \qquad (1)$$

Considering that it is impossible to calculate $\vec{v}_1$ when the dot first appears, we only analyze the remaining $K-1$ positions. In the following $K-1$ positions, if the capture rate is higher than H, we believe that the recorded data is indeed from a live user.

E. Identity Verifier

The goal of this module is to determine whether the recorded data is consistent with the identity the user declared. We begin by introducing the biometric features extracted, followed by an explanation of the classification model employed.

1) Hand-engineered Feature Extraction: We discover that despite the overall gaze trajectory is random, these sub-trajectories can exhibit certain stability (as shown in Figure 6, each containing acceleration and deceleration two phases). An authentication attempt contains multiple sub-trajectories, each of which allows us to compute a set of features.

For each sub-trajectory, we extract features from $\dot{\Theta}^s$ and $\ddot{\Theta}^s$, where $\dot{\Theta}^s$ and $\ddot{\Theta}^s$ are the sub-trajectory's gaze angular speed and acceleration. We extract seven statistical features, including mean, maximum, standard deviation, relative standard deviation, sum of absolute differences, absolute energy, and autocorrelation. These features are widely used for time series analysis [15], [16], [17]. To increase robustness, the final feature values are computed as the mean of individual sub-trajectory features.

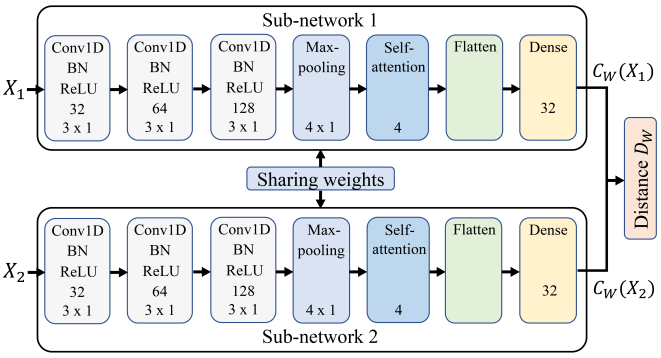


Fig. 8. Siamese network architecture.

2) SASN-based Feature Extraction: Besides the hand-engineered features, we also design a Self-Attention-based Siamese Network (SASN) to extract dot position-irrelevant features. During training, the Siamese network utilizes twin sub-networks with the same architecture and weights to compute a distance metric of two inputs (i.e., $\dot{\Theta}$ of two samples) [18], [19]. The unique structure of the Siamese network enables it to achieve dot position-irrelevant feature extraction by employing a unique training sample selection scheme.

Figure 8 displays the SASN architecture in EyeAuth. Each sub-network starts with three convolutional layers with 3×1 kernels for learning local features, followed by a max-pooling layer with kernel size 4×1. A stride of 1 is employed. A batch normalization layer and a ReLU active function are added after each convolution layer. Subsequently, a self-attention layer with four heads is incorporated for learning global features. Finally, a flatten layer and a dense layer are utilized to reduce the feature dimension. Given a pair of samples, the network outputs two fixed-size features, $C_W(X)$, and computes their distance, D_W, to quantify similarity.

The loss function, denoted as $L(W)$, is formulated to optimize the parameters W of each sub-network.

$$L(W) = \sum_{i=1}^{N} Y(D_W^i)^2 + (1-Y)max(M - D_W^i, 0)^2 \qquad (2)$$

Label / OTGC code	Same	Different
Same	×	√
Different	√	×

Fig. 9. Illustration of SASN training samples selection.

Here, Y is an indicator to show if the two input samples are from the same user, i.e., if they are from the same user (i.e., positive pair), $Y = 1$, otherwise (i.e., negative pair), $Y = 0$. N is the batch size and M is the margin representing the decrease interval. The designed SASN aims at minimizing the loss $L(W)$, thereby reducing the distance for positive pairs while increasing the distance for negative pairs.

Figure 9 illustrates our training sample selection scheme. Specifically, EyeAuth classifies each pair of samples into one of four classes based on their OTGC codes and user labels. If the user labels are the same (positive pairs), EyeAuth selects sample pairs with different OTGC codes, so that the SASN can learn to ignore dot position differences from the same user. If the user labels are different (negative pairs), EyeAuth selects sample pairs with the same OTGC code, so that the SASN can distinguish samples from different users based on the inherent biometrics. After training, the sub-network is used for feature extraction.

We combine the hand-engineered features and SASN-based features, then utilize t-SNE projection [20] to visualize the resulting feature space. t-SNE can map high-dimensional features to lower dimensions while preserving the relative distance information between samples. As shown in Figure 7, the samples from different subjects are clustered separately, validating the distinguishability of our extracted features under random gaze challenges.

3) Classification: We employ four different classifiers, including Naive Bayes (NB), K-Nearest Neighbor (KNN), Decision Tree (DT), and Support Vector Machine (SVM): i) NB [21] is a classifier grounded in Bayesian theory, classifying a sample into the class with the highest posterior probability; ii) KNN [22] is a density-based classifier that determines the class of a given sample by considering the classes of its K nearest neighbors; iii) DT [23] is a tree-structured classifier. It is essentially a recursive function where each path from the root node to a leaf node represents a decision rule, enabling classification based on the sample features; iv) SVM [24] is a distance-based classifier. It works by mapping sample features into a high-dimensional space with kernel function, facilitating feature linear separability. A grid search is performed to find the optimal parameter combination of these classifiers.

IV. Implementation

A. Experimental Setup

For data capturer, the OTGC code length $K = 25$. The dot has a red color, with a time interval of 500ms to change position. The camera sampling rate $f_s = 30Hz$. For data preprocessor, the S-G filter window width is set to $W = 5$. To detect sub-trajectory, we set $T_b = 50°/s$, $T_h = 15°/s$, $T_l = 10°/s$, and $L = 5$. For liveness verifier, we employ grid search to determine the optimal α and H values. Our goal is to achieve an ideal balance between rejecting replay attack samples and accepting legitimate ones. As a result, we set angle $\alpha = 30°$ and threshold $H = 35\%$ (see Sec. V-C3), which are applied across all subjects. For identity verifier, the dimension of hand-engineered feature and SCSN-based feature are 14 (7 features from both $\dot{\Theta}^s$ and $\ddot{\Theta}^s$) and 32 (output of the dense layer), respectively.

B. Data Collection

After receiving the IRB approval from our university, we start recruiting subjects for data collection. We employ 40 subjects (25 males and 15 females) aged from 19 to 29 from our university. 22 subjects wear glasses during data collection. All the subjects are explained the research purpose, the data collection process, and the measures taken to protect their personal information.

We collect data on a Samsung Galaxy S10. Before data collection, subjects need to observe the gaze challenge several times to get familiar with it. In each session, the gaze challenge is displayed based on 40 different OTGC codes. Each subject participates in five sessions, resting for three minutes between two sessions to simulate putting down their smartphones for a break. This generates a *basic dataset* of $40 \times 5 \times 40 = 8000$ samples.

We also collect datasets under different conditions such as shooting distances, camera angles, using environments, etc. For each condition (e.g., 30cm shooting distance), data from five subjects are collected in one session. We select subjects from the basic dataset to train user-specific models, and conduct testing with the user data under different conditions.

C. Evaluation Metrics

The following metrics are utilized for evaluation. True Acceptance Rate (TAR) is the ratio of legitimate users gaining access. True Rejection Rate (TRR) is the ratio of non-users being denied access. Balanced Accuracy (BAC) is used to evaluate the model's overall performance under unbalanced data [25]. It is defined as the average of the TAR and TRR.

V. Evaluation

A. Overall Performance

We use the *basic dataset* to evaluate the system's overall performance. Each subject is respectively chosen as the user and the others as non-users. 10-fold cross-validation is performed for evaluation.

1) Different Feature Sets: To examine the security gain brought by integrating hand-engineered and SASN-based features, we compare the authentication performance achieved by applying different feature sets. SVM serves as the classifier for this evaluation. Figure 10 presents the comparison of BAC among 15 randomly selected subjects. We then test the performance of different feature sets across 40 subjects. Overall, when using hand-engineered features alone, our system achieves an average BAC of 88.75%, TAR of 9.26%,

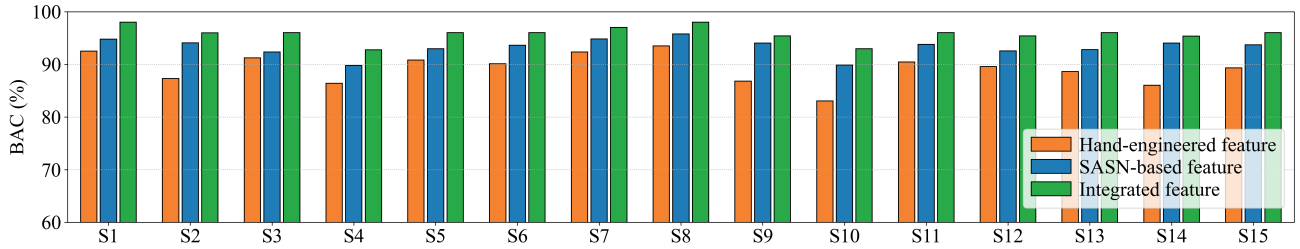


Fig. 10. System performance across different subjects.

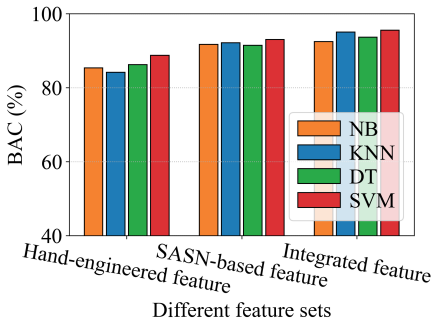


Fig. 11. BAC of different classifiers.

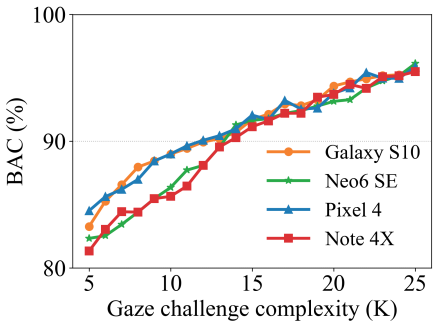


Fig. 12. Impact of gaze challenge complexity.

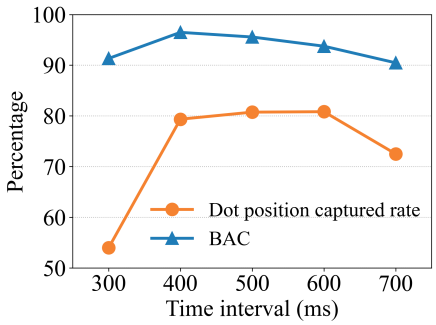


Fig. 13. Impact of gaze challenge time interval.

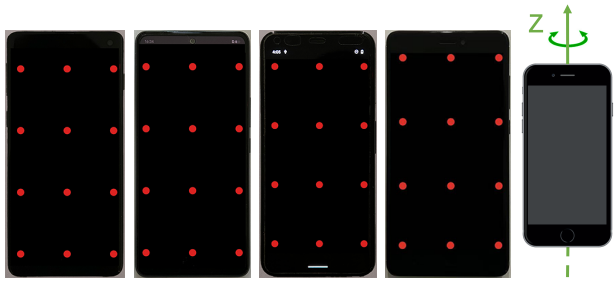


Fig. 14. Different devices employed: Samsung Galaxy S10 (a), IQOO Neo6 SE (b), Google Pixel 4 (c), and Redmi Note 4X (d); the method of device rotation (e).

and TRR of 14.57%. Using SASN-based features achieves an average BAC of 93.03%, TAR of 6.18%, and TRR of 9.14%. Combining these features improves the authentication performance, resulting in a BAC of 95.56%, TAR of 3.48%, and TRR of 5.12%. This implies that the two types of features complement each other, leading to additional security gains.

2) Different Classifiers: We compare the system performance under different classifiers, as depicted in Figure 11. We observe variations in the performance of different classifiers. Specifically, when using hand-engineered features alone, the average BACs for NB, KNN, DT, and SVM are 85.36%, 84.18%, 86.28%, and 88.75%, respectively. Utilizing SASN-based features leads to average BACs of 91.76%, 92.15%, 91.47%, and 93.03%. Incorporating both feature sets improves the BACs to 92.49%, 95.06%, 93.69%, and 95.56%, respectively. Overall, the performance of SVM is slightly superior to the other three classifiers. Therefore, in the subsequent experiments, SVM is considered the default classifier.

B. Impact of Various Factors

1) Gaze Challenge Complexity: Gaze challenge complexity is determined by the parameter K in one authentication attempt. Specifically, each sample has an OTGC code length of 25. If $K < 25$, the actual eye movement should correspond to the period before the dot appears at the $(K+1)_{th}$ position. By detecting the dot appearance time, we obtain the eye movement data corresponding to different gaze challenge complexity. Four different smartphones are utilized in this experiment: 1) Samsung Galaxy S10 features a 6.1-inch screen and a 10-megapixel front camera; 2) IQOO Neo6 SE features a 6.62-inch screen and a 16-megapixel front camera; 3) Google Pixel 4 features a 5.7-inch screen and an 8-megapixel front camera; 4) Redmi Note 4X features a 5.5-inch screen and a 5-megapixel front camera. Figure 14(a)-(d) show the devices employed, with all possible red dot positions displayed on each screen. For Samsung Galaxy S10, *basic dataset* is utilized for evaluation. For the other three smartphones, data collection involves five participants, each contributing data across five sessions per device. The model is trained and tested on data from the same device using 10-fold cross-validation. Figure 12 presents the results. We observe that as K increases, the BAC increases across four devices. In particular, when $K = 16$, the BACs are 91.74%, 92.18%, 92.06%, and 91.61% on Samsung Galaxy S10, IQOO Neo6 SE, Google Pixel 4, and Redmi Note 4X, respectively. When $K = 25$, the BACs on four devices increase to 95.56%, 96.12%, 95.85%, and 95.49%, respectively.

2) Gaze Challenge Time Interval: The time interval T is the duration between dot position changes. When T is too small, a user's eyes may not be able to follow the dot timely. When T is too large, the gaze challenge cannot sufficiently

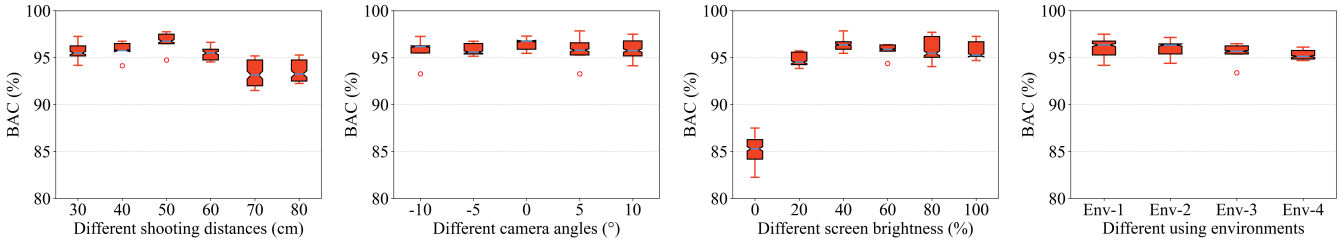


Fig. 15. Impact of different shooting distances (a), camera angles (b), screen brightnesses (c), and using environments (d).

trigger the user's reflexive eye movements. Figure 13 presents the effect of T. For BAC, as T increases from 300ms to 700ms, it first increases and then decreases gradually. When T=300ms, rapid dot position changes making it hard for eyes to follow, resulting in a low BAC. When T=400ms, the BAC reaches 96.48%, as user reflexive eye movements are fully triggered. After that, with the increase of T, BAC decreases gradually. This is because longer intervals lead to more voluntary eye movements, usually unstable [4]. For the dot position captured rate, it remains around 80% at intervals of T=400, 500, and 600ms, indicating that human eyes can effectively track dot position changes at these intervals. When T=700ms, the capture rate decreases to 72.28%, suggesting that a long interval leads to the user being absent-minded, and therefore cannot follow dot position change timely.

3) Shooting Distances: To evaluate the impact of shooting distances, we consider distances ranging from 30cm to 80cm at intervals of 10cm. Figure 15(a) presents the results. We observe that when shooting at a long distance, there is a slight decrease in the BAC. Specifically, the BACs are 95.67%, 95.77%, 96.64%, 95.46%, 93.31%, and 93.61% when the distances are set to 30cm, 40cm, 50cm, 60cm, 70cm, and 80cm, respectively. The results confirm our system can work well across a wide range of distances. This is because the gaze estimation algorithm [10] employed is based on deep learning, which is robust across various distances.

4) Camera Angles: To evaluate the impact of camera angles, we consider both positive (rotate the smartphone clockwise) and negative (rotate the smartphone counterclockwise) shooting angles. The device rotates around the z-axis as shown in Figure 14(e). We consider a maximum rotation angle of $10°$, where the subject's face is near the edge of the captured image. As shown in Figure 15(b), the BACs are 95.69%, 95.87%, 96.43%, 95.75%, and 95.87% when the shooting angles are $-10°$, $-5°$, $0°$, $5°$, and $10°$, respectively. The results show that EyeAuth can perform well across different shooting angles.

5) Screen Brightness: To evaluate the impact of screen brightness, we consider six brightness levels distributed from 0% to 100%. As depicted in Figure 15(c), the BACs are 85.09%, 94.78%, 96.45%, 95.73%, 95.89%, and 95.80% when the brightness are 0%, 20%, 40%, 60%, 80%, and 100%, respectively. We observe that the BAC is high in most brightness conditions, demonstrating our system can work well around a wide range of screen brightness. At 0% brightness, the BAC decreases significantly. This is because the excessively low screen brightness makes it difficult for the user to perceive dot position changes, resulting in insufficient reflexive eye movements.

6) Using Environments: The smartphone could be used in different environments. We test the performance of EyeAuth under four types of environments: *Env-1* is an office with lights turned on; *Env-2* is a park with natural sunlight; *Env-3* is a bedroom with lights turned off and curtains closed; *Env-4* is a roadside with street lamps at night. We train our model with data from *Env-1* and test it across four environments. As shown in Figure 15(d), the BACs are 96.01%, 95.94%, 95.44%, and 95.31% for environments *Env-1* to *Env-4*. This demonstrates that as long as the lighting meets the basic requirements for image recognition, the system can exhibit stable performance.

7) Consistency Over Time: To verify the long-term consistency of our system, we conduct a four-week study. We recruit five subjects for this experiment. Each week, we gather data across five sessions, with each session containing 40 different OTGC codes. The first week's data is drawn from the user's *basic dataset*. We use 80% of the samples from week 1 to train our model and test it with data from weeks 1, 2, 3, and 4. Specifically, the average BACs are 96.58%, 96.17%, 95.94%, and 96.74% in weeks 1, 2, 3, and 4, respectively. We do not find an obvious BAC decrease within the four weeks, which is attributed to the good consistency of reflexive eye movements.

8) Wearing Glasses: To evaluate the impact of wearing glasses, we divide the recruited subjects in the *basic dataset* into two groups: group-A comprises 22 subjects wearing glasses and group-B comprises 18 subjects not wearing glasses. We assess the system performance within each group. The result is shown in Figure 16. Specifically, when subjects wear glasses, the average BACs are 84.29%, 91.76%, and 93.19% for hand-engineered, SASN-based, and integrated features, respectively. When subjects do not wear glasses, the BACs increase to 89.19%, 93.86%, and 96.48%, respectively. The results demonstrate that wearing glasses has a slight impact on the system performance.

C. Security Analysis

We evaluate the security of our proposed authentication system under the three types of attacks.

1) Blind Attack: We evaluate our system under blind attackers, who try to gain access to the system with random observation. We recruit seven subjects as the attacker, with each completing a two-session data collection. The remaining subjects in *basic dataset* are treated as victims. As a result, our system could defend against 97.38% blind attacks. The result shows that our method guarantees security under blink attacks.

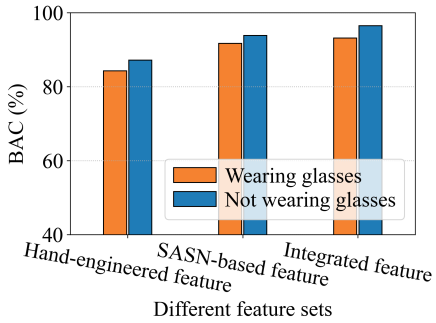


Fig. 16. Impact of wearing glasses.

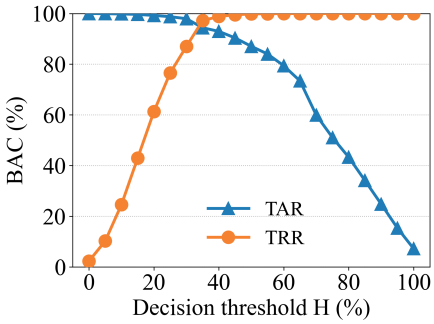


Fig. 17. Evaluation of replay attack.

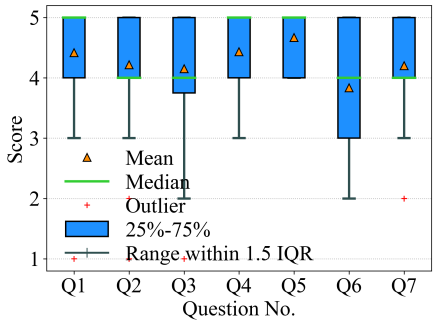


Fig. 18. Questionnaire score.

2) Impersonation Attack: To evaluate the system performance under impersonation attacks, we recruit seven subjects to act as attackers. Each attacker is required to observe the authentication videos (recorded by the smartphone front camera) of three victims. After being confident about their observations, attackers need to mimic the behavior of each target victim and collect data for one session. The experiment shows that our system could defend 95.81% of impersonate attacks. The reason lies in eye movements being complex and presented on a small scale, thus hard to mimic.

3) Replay Attack: We evaluate the system performance under two kinds of replay attacks. For *video display attack*, 5 subjects act as victims. We randomly select 100 samples from each of their datasets and replay them to the target device. Given the huge pool of OTGC codes, we assume that the code displayed on the target device differs from the video's OTGC code. For *video injection attack*, we consider all 40 subjects as victims. Each session in the *basic dataset* contains 40 different OTGC codes. Since the OTGC code is unique for each authentication, this enables us to perform $40 \times 39 \times 5 = 7800$ replay attacks by cross-matching different gaze challenges and eye movement responses for each subject. The final evaluation is performed on $7800 \times 40 = 312000$ replay attacks.

For *video injection attack*, the TRR is 98.46%, suggesting that EyeAuth can effectively resist this replay attack. For *video injection attack*, Figure 17 presents the relationship between TAR and TRR under different decision thresholds H. We observe that with H increases, it is easier to detect replay attacks (TRR increase), but the legitimate sample is more likely to be misclassified as a replay (TAR decrease). The two curves cross near H=35%. At this threshold, TRR=97.39% and TAR=94.18%. This demonstrates that our system can effectively resist *video injection attack* while reliably authenticating legitimate users.

D. User Study

We assess the usability of our system from the subjects' perspective. After data collection, subjects are asked to complete a questionnaire by responding to 7 questions on a 5-point Likert scale (with 1 = strongly disagree and 5 = strongly agree). The questions are listed in Table I. All 40 subjects respond to the questions, with Figure 18 presenting the results. The average score for different questions are: Q1 ($\mu = 4.41, \sigma = 0.78$), Q2 ($\mu = 4.21, \sigma = 0.95$), Q3 ($\mu = 4.15, \sigma = 0.99$), Q4 ($\mu = 4.43, \sigma = 0.69$), Q5 ($\mu = 4.66, \sigma = 0.47$), Q6 ($\mu = 3.83, \sigma = 1.08$), and Q7 ($\mu = 4.20, \sigma = 0.81$). Overall, most subjects express their satisfaction with EyeAuth, as evidenced by a median score of 5 (Q1, Q4, Q5) or 4 (Q2, Q3, Q6, Q7) across seven questions.

TABLE I
CONTENT OF QUESTIONNAIRE.

No.	Questions
Q1	EyeAuth is a secure authentication scheme.
Q2	EyeAuth is easy to use.
Q3	There is no discomfort using EyeAuth.
Q4	EyeAuth is easy to learn.
Q5	EyeAuth does not introduce much cognitive load .
Q6	The authentication time is acceptable.
Q7	I am willing to use EyeAuth as an alternative method on my devices.

E. Comparision with Related Work

We implement the method proposed by EyeVeri [8] to explore the impact of replay attacks. EyeVeri presents four visual stimuli on a smartphone as gaze challenges: Fruit-Row (active scanning), Corner-Gif (passive following), Illusion-Image (strong stimulation), and Simple-Dot (weak stimulation). Gaze angles (horizontal and vertical) are estimated from the video captured by the front camera, from which EyeVeri extracts various statistical features to classify users using an SVM. We re-implement EyeVeri's method, with design details in Table II. Five subjects participate in the study, each contributing 200 samples for each visual stimulus. We train the SVM using 80% of the randomly selected samples from each visual stimulus and test the *video display attack* on the remaining 20%. This process is repeated multiple times for reliable evaluation. As a result, the TRRs are 14.56%, 36.75%, 22.69%, and 18.74% for Fruit-Row, Corner-Gif, Illusion-Image, and Simple-Dot, respectively. In contrast, our system achieves a TRR of 98.46% for the *video display attack*. The inherent reason is that EyeVeri's visual stimuli lack randomness, resulting in similar gaze patterns across different trials, making it vulnerable to replay attacks.

VI. LIMITATIONS

Although we make great efforts to maintain our studies' validity, there are limitations in our study and experiments.

TABLE II
DETAILS OF RE-IMPLEMENTING EYEVERI.

	Device	Visual stimuli	Sensor	Gaze data	Feature	Classifier
EyeVeri	Smartphone	Three pictures and a GIF	Front camera	Gaze angles (horizontal and vertical)	Statistical features	SVM
Re-implementation	Smartphone	Three pictures and a GIF	Front camera	Gaze angles (Θ_x and Θ_y)	Statistical features	SVM

EyeAuth detects eye movements based on the front camera, which has similar limitations to face recognition. For example, similar to face recognition, eye movement detection relies on a clear image capture, which is not suitable under poor lighting, off-normal shooting angles, and dynamic environments (e.g., running), which may cause more false rejections. There exists a trade-off between security and usability. Specifically, increasing the complexity of the gaze challenge improves authentication accuracy by capturing more eye movement data, but it also extends the authentication time required. Conversely, simplifying the challenge reduces time but compromises performance. To balance this, EyeAuth can integrate with existing authentication systems, offering liveness verification with low gaze complexity while maintaining authentication performance.

VII. RELATED WORK

In this section, we introduce the user authentication methods that are deployed on smartphones. In addition, we compare existing eye-based authentication methods and our method.

A. Smartphone-based Authentication.

To protect sensitive information on smartphones, two types of authentication methods are widely deployed. Knowledge-based methods, such as PINs, passwords, and lock patterns [26], can be easily observed by others in proximity and are vulnerable to shoulder-surfing attacks. Biometric-based methods, using physiological traits like fingerprints, facial features, palm prints, or behavioral patterns, seem to be usable and convenient for smartphone authentication.

Fingerprint-based authentication [36] is popular in smartphones for its convenience. However, fingerprint-based systems are susceptible to spoofing attacks [28], where attackers deceive the system with fake materials, like gummy fingers that have fingerprint impressions or human-based instruments [37]. Face authentication, like FaceID [29], is another widely used smartphone biometric. However, it is vulnerable to spoofing techniques like face morphing [38], [39], and struggles with reliable identification when users wear masks, reducing its effectiveness. Simple palmprint authentication can be implemented with a camera by capturing skin patterns like lines, points, and texture [40], [41]. However, this method is less reliable and prone to spoofing attacks, as it can be tricked with simple images [37]. Some works [42], [30] use acoustic waves to sense palms, which could distinguish real palms and images effectively. But they often face usability issues, such as being sensitive to hand gestures and sensing distance. Some authentication methods utilize behavioral patterns, such as voices [32], gaits [33], signatures [34], and breaths [43], [35], [44]. Among these solutions, voice-based user authentication is widely deployed in smartphones. However, recent studies reveal that voice biometrics are prone to speech synthesis attacks [45]. These breathing activities, typically captured with novel wireless sensing methods, such as sound [43], RFID [35], and WiFi [44], have demonstrated performance in identifying users. However, these wireless authentication methods are generally hindered by inconsistencies in user behavior, as well as the effects of sensing distances and angles, limiting their reliability in real-world applications. Compared with these biometrics, eye movements are complex and presented on a miniature scale, thus highly immune to accurate artificial creation of adversaries [46]. By incorporating random visual stimuli, our approach is further enhanced in defending against replay attacks. A comprehensive comparison with prior user authentication schemes for smartphones is provided in Table III.

B. Eye-based Authentication.

Existing eye-based authentication methods could be divided into two categories: knowledge-based and biometric-based.

For knowledge-based authentication, users are required to remember a "password" in advance and use eye movement as the input method. The inputs could be fixations [47], [48], [49], [50], [51], gaze gestures [52], [53], [54], [55], [27], [56], and smooth pursuits [57], [58] according to eye movement patterns. Although these methods prevent attackers from obtaining passwords through shoulder surfing, it is still possible to infer passwords by comparing the user's eye movements with screen stimuli.

Biometric-based authentication utilizes eye physiological or behavioral information to authenticate users. For example, OcuLock [59] leverages electrooculography to measure periocular electric voltage variations during eye activities for authentication. Zhang et al. [5] built a prototype smart glasses, allowing capturing individual eye movements when they observe app contents for authentication. But these methods require additional hardware thus not suitable for smartphones. Blinkey [60] utilizes the unique pupil size variation during blinking for authentication. SoundLock [61] identifies users by analyzing pupillary responses to auditory stimuli. Sluganovic et al. [4] employ an eye tracker as the data acquisition device, analyzing the eye movement response of random dots to authenticate users. Eberz et al. [6] analyze publicly available eye-tracking data, investigating the feasibility of transferring eye movement data in different authentication tasks. Lohr et al. [7] propose a DenseNet-based architecture for eye movement biometrics extraction with the high-quality eye-tracking dataset and verify its efficacy of deploying on VR/AR devices. However, these methods all require eye trackers [60], [61], [4], [6], [7] for data acquisition, which is expensive and unavailable on mobile devices. Zheng et al. [31] employ a

TABLE III
COMPARISON AMONG DIFFERENT USER AUTHENTICATION APPROACHES FOR SMARTPHONES.

Work	Protocol	Feature	Extra sensor-free	Performance	Login time	Against replay	Against shoulder-surfing	Against synthesis
PIN	Knowledge	N/A	●	N/A	$< 2s$	○	○	N/A
Password	Knowledge	N/A	●	N/A	$< 2s$	○	○	N/A
Lock pattern [26]	Knowledge	N/A	●	N/A	$< 2s$	○	○	N/A
GTmoPass [27]	Knowledge	Eye movement	●	N/A	$3.9s$	○	●	N/A
Sousedik et al. [28]	Physiological	Fingerprint	●	$< 1\% EER$	$< 1s$	○	●	○
FaceID [29]	Physiological	Face	●	$< 1\% EER$	$< 1s$	○	●	○
Wu et al. [30]	Physiological	Palmprint	●	$2.45\% EER$	$\sim 2s$	○	●	○
Zheng et al. [31]	Physiological	Periocular	●	$98.93\% TPR$	$1.88s$	●	●	●
Voicelive [32]	Behavioral	Voice	●	$96.11\% Acc.$	$\sim 7.5s$	●	●	○
Ren et al. [33]	Behavioral	Gait	●	$9\% FPR$	$\sim 10s$	○	◐	◐
Ren et al. [34]	Behavioral	Signature	●	$2\% EER$	$\sim 20s$	○	◐	◐
Ning et al. [35]	Behavioral	Breath	○	$97\% Acc.$	$> 10s$	●	●	●
EyeVeri [8]	Behavioral	Eye movement	●	$88.73\% BAC$	$\sim 10s$	○	●	◐
EyeAuth (this work)	Behavioral	Eye movement	●	$95.56\% BAC$	$\sim 10s$	●	●	●

●: method fulfills criterion. ◐: method quasi-fulfills criterion. ○: method does not fulfill criterion. N/A: not enough information.

smartphone camera to capture periocular information during eye movement for authentication. EyeVeri [8] employs four visual stimuli displayed on a smartphone, named Fruit-Row (active scanning), Corner-Gif (passive following), Illusion-Image (strong stimulation), and Simple-Dot (weak stimulation) to trigger eye movements. However, EyeVeri is vulnerable to replay attacks because it relies on static visual stimuli. Moreover, it suffers long-term performance degradation as users become familiar with these static images.

Different from these studies, EyeAuth leverages solely eye movement biometrics to authenticate users and can be deployed on commodity mobile devices. Fortified with random gaze challenge, EyeAuth can efficiently resist replay attacks and prevent changes in gaze response due to habituation.

VIII. CONCLUSION

In this paper, we present EyeAuth, a replay-resistant eye movement authentication method for commodity smartphones. It employs built-in hardware on off-the-shelf devices, including the screen and camera. To mitigate the threats of replay attacks, it uses a dot that changes position multiple times as the gaze challenge and characterizes reflexive eye movements with hand-engineered and SASN-based features. We recruit 40 subjects to evaluate the reliability and security of EyeAuth. The results demonstrate that EyeAuth can achieve a commendable BAC of 95.56% while defeating impersonate and replay attacks.

ACKNOWLEDGMENTS

This research was supported in part by the National Key R&D Program of China under grant No. 2021YFB2700200, the National Natural Science Foundation of China under grants No. 62172303, the Key R&D Program of Hubei Province under grant No. 2024BAB018, and the Key R&D Program of Shandong Province under grant No. 2022CXPT055. The corresponding author is Jing Chen.

REFERENCES

[1] Data leak exposes unchangeable biometric data of over 1 million people, 2019

[2] German R L, Barber K S. Consumer attitudes about biometric authentication. Univ. Texas, Austin, TX, USA, UT CID Rep, 2018, 18–03

[3] Griffith H, Lohr D, Abdulin E, Komogortsev O. Gazebase, a large-scale, multi-stimulus, longitudinal eye movement dataset. Scientific Data, 2021, 8(1): 184

[4] Sluganovic I, Roeschlin M, Rasmussen K B, Martinovic I. Using reflexive eye movements for fast challenge-response authentication. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security (CCS). 2016, 1056–1067

[5] Zhang Y, Hu W, Xu W, Chou C T, Hu J. Continuous authentication using eye movement response of implicit visual stimuli. In: Acm on interactive, mobile, wearable and ubiquitous technologies (IMUWT). 2018

[6] Eberz S, Lovisotto G, Rasmussen K B, Lenders V, Martinovic I. 28 blinks later: Tackling practical challenges of eye movement biometrics. In: ACM SIGSAC Conference on Computer and Communications Security (CCS). 2019

[7] Lohr D, Komogortsev O V. Eye know you too: Toward viable end-to-end eye movement biometrics for user authentication. In: IEEE Transactions on Information Forensics and Security (TIFS). 2022

[8] Song C, Wang A, Ren K, Xu W. Eyeveri: A secure and usable approach for smartphone user authentication. In: IEEE Conference on Computer Communications (INFOCOM). 2016

[9] Sluganovic I, Roeschlin M, Rasmussen K B, Martinovic I. Analysis of reflexive eye movements for fast replay-resistant biometric authentication. ACM Transactions on Privacy and Security (TOPS), 2018, 22(1): 1–30

[10] Zhang X, Sugano Y, Bulling A. Evaluation of appearance-based methods and implications for gaze-based applications. In: Proceedings of CHI conference on human factors in computing systems (CHI). 2019, 1–13

[11] Smartphone unit shipments worldwide by screen size from 2018 to 2022, 2021

[12] Savitzky A, Golay M J. Smoothing and differentiation of data by simplified least squares procedures. Analytical chemistry, 1964, 36(8): 1627–1639

[13] Holmqvist K, Nyström M, Andersson R, Dewhurst R, Jarodzka H, Weijer V. d J. Eye tracking: A comprehensive guide to methods and measures. oup Oxford, 2011

[14] Vartiainen J, Lehtomaki J, Saarnisaari H. Double-threshold based narrowband signal extraction. In: IEEE vehicular technology conference. 2005

[15] Wu C, He K, Chen J, Zhao Z, Du R. Liveness is not enough: Enhancing fingerprint authentication with behavioral biometrics to defeat puppet attacks. In: USENIX Security Symposium (USENIX). 2020

[16] Chen Y, Jin X, Sun J, Zhang R, Zhang Y. Powerful: Mobile app fingerprinting via power analysis. In: IEEE Conference on Computer Communications (INFOCOM). 2017

[17] Li Y, Hu H, Zhou G. Using data augmentation in continuous authentication on smartphones. In: IEEE Internet of Things Journal (IoT-J). 2018

[18] Koch G, Zemel R, Salakhutdinov R, others . Siamese neural networks for one-shot image recognition. In: International Conference on Machine Learning. 2015

[19] Fereidooni H, König J, Rieger P, Chilese M, Gökbakan B, Finke M, Dmitrienko A, Sadeghi A R. Authentisense: A scalable behavioral biometrics authentication scheme using few-shot learning for mobile platforms. In: Network and Distributed System Security Symposium (NDSS). 2023

[20] Maaten V. d L, Hinton G. Visualizing data using t-sne. Journal of machine learning research, 2008

[21] Berrar D. Bayes' theorem and naive bayes classifier., 2019

[22] Cunningham P, Delany S J. K-nearest neighbour classifiers-a tutorial. In: ACM computing surveys. 2021

[23] Song Y Y, Ying L. Decision tree methods: applications for classification and prediction. In: Shanghai archives of psychiatry. 2015

[24] Auria L, Moro R A. Support vector machines (svm) as a technique for solvency analysis. In: DIW Berlin discussion paper. 2008

[25] Brodersen K H, Ong C S, Stephan K E, Buhmann J M. The balanced accuracy and its posterior distribution. In: International conference on pattern recognition (ICPR). 2010

[26] Uellenbeck S, Dürmuth M, Wolf C, Holz T. Quantifying the security of graphical passwords: The case of android unlock patterns. In: ACM conference on Computer and communications security (CCS). 2013

[27] Khamis M, Hasholzner R, Bulling A, Alt F. Gtmopass: two-factor authentication on public displays using gaze-touch passwords and personal mobile devices. In: ACM International Symposium on Pervasive Displays. 2017

[28] Sousedik C, Busch C. Presentation attack detection methods for fingerprint recognition systems: a survey. Iet Biometrics, 2014

[29] Face id security, 2024

[30] Wu C, Chen J, He K, Zhao Z, Du R, Zhang C. Echohand: High accuracy and presentation attack resistant hand authentication on commodity mobile devices. In: ACM conference on computer and communications security (CCS). 2022

[31] Zheng Z, Wang Q, Wang C, Zhou M, Zhao Y, Li Q, Shen C. Where are the dots: Hardening face authentication on smartphones with unforgeable eye movement patterns. In: IEEE Transactions on Information Forensics and Security (TIFS). 2022

[32] Zhang L, Tan S, Yang J, Chen Y. Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones. In: ACM Conference on Computer and Communications Security (CCS). 2016

[33] Ren Y, Chen Y, Chuah M C, Yang J. User verification leveraging gait recognition for smartphone enabled mobile healthcare systems. IEEE Transactions on Mobile Computing (TMC), 2014

[34] Ren Y, Wang C, Chen Y, Chuah M C, Yang J. Signature verification using critical segments for securing mobile transactions. IEEE Transactions on Mobile Computing (TMC), 2019

[35] Ning J, Xie L, Wang C, Bu Y, Xu F, Zhou D W, Lu S, Ye B. Rf-badge: Vital sign-based authentication via rfid tag array on badges. IEEE Transactions on Mobile Computing (TMC), 2021

[36] Maltoni D, Maio D, Jain A K, Prabhakar S, others . Handbook of fingerprint recognition. Springer, 2009

[37] 30107-1 I. Information Technology: Biometric Presentation Attack Detection. ISO/IEC, 2016

[38] Ferrara M, Franco A, Maltoni D. On the effects of image alterations on face recognition accuracy. Face recognition across the imaging spectrum, 2016

[39] Scherhag U, Rathgeb C, Merkle J, Breithaupt R, Busch C. Face recognition systems under morphing attacks: A survey. IEEE Access, 2019

[40] Bhilare S, Kanhangad V, Chaudhari N. A study on vulnerability and presentation attack detection in palmprint verification system. Pattern Analysis and Applications

[41] Jia W, Zhang B, Lu J, Zhu Y, Zhao Y, Zuo W, Ling H. Palmprint recognition based on complete direction representation. IEEE Transactions on Image Processing (TIP), 2017

[42] Wang L, Chen W, Jing N, Chang Z, Li B, Liu W. Acopalm: Acoustical palmprint-based noncontact identity authentication. IEEE Transactions on Industrial Informatics, 2022

[43] Chen Y, Xue M, Zhang J, Guan Q, Wang Z, Zhang Q, Wang W. Chestlive: Fortifying voice-based authentication with chest motion biometric on smart devices. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMUWT), 2021

[44] Huang P, Zhang D, Geng R, Chen Y. Continuous user authentication using wifi. In: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). 2022

[45] Kinnunen T, Wu Z, Nicholas Evans E, Yamagishi J. Automatic speaker verification spoofing and countermeasures challenge (asvspoof 2015) database. 2018

[46] Komogortsev O V, Karpov A, Holland C D. Attack of mechanical replicas: Liveness detection with eye movements. IEEE Transactions on Information Forensics and Security (TIFS), 2015

[47] Kumar M, Garfinkel T, Boneh D, Winograd T. Reducing shoulder-surfing by using gaze-based password entry. In: Symposium On Usable Privacy and Security (SOUPS). 2007

[48] Forget A, Chiasson S, Biddle R. Shoulder-surfing resistance with eye-gaze entry in cued-recall graphical passwords. In: ACM Conference on Human Factors in Computing Systems (CHI). 2010

[49] Bulling A, Alt F, Schmidt A. Increasing the security of gaze-based cued-recall graphical passwords using saliency masks. In: ACM Conference on Human Factors in Computing Systems (CHI). 2012

[50] Hoanca B, Mock K. Secure graphical password system for high traffic public areas. In: ACM symposium on Eye tracking research and applications. 2006

[51] Dunphy P, Fitch A, Olivier P. Gaze-contingent passwords at the atm. In: ACM Conference on Communication by Gaze Interaction – Communication, Environment and Mobility Control by Gaze. 2008

[52] De Luca A, Weiss R, Drewes H. Evaluation of eye-gaze interaction methods for security enhanced pin-entry. In: Australasian conference on Computer-Human Interaction: Entertaining User Interfaces. 2007

[53] De Luca A, Weiss R, Hussmann H, An X. Eyepass-eye-stroke authentication for public terminals. In: ACM Extended Abstracts on Human Factors in Computing Systems (CHI EA). 2008

[54] Bhatti O S, Barz M, Sonntag D. Eyelogin-calibration-free authentication method for public displays using eye gaze. In: ACM symposium on Eye tracking research and applications. 2021

[55] Khamis M, Alt F, Hassib M, Zezschwitz v E, Hasholzner R, Bulling A. Gazetouchpass: Multimodal authentication using gaze and touch on mobile devices. In: ACM Extended Abstracts on Human Factors in Computing Systems (CHI EA). 2016

[56] Khamis M, Hassib M, Zezschwitz E v, Bulling A, Alt F. Gazetouchpin: protecting sensitive data on mobile devices using secure multimodal authentication. In: ACM International Conference on Multimodal Interaction (ICML). 2017

[57] Khamis M, Trotter L, Mäkelä V, Zezschwitz E v, Le J, Bulling A, Alt F. Cueauth: Comparing touch, mid-air gestures, and gaze for cue-based authentication on situated displays. In: ACM on interactive, mobile, wearable and ubiquitous technologies (IMUWT). 2018

[58] Liu D, Dong B, Gao X, Wang H. Exploiting eye tracking for smartphone authentication. In: Applied Cryptography and Network Security (ACNS). 2015

[59] Luo S, Nguyen A, Song C, Lin F, Xu W, Yan Z. Oculock: Exploring human visual system for authentication in virtual reality head-mounted display. In: Network and Distributed System Security Symposium (NDSS). 2020

[60] Zhu H, Jin W, Xiao M, Murali S, Li M. Blinkey: A two-factor user authentication method for virtual reality devices. In: ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT). 2020

[61] Zhu H, Xiao M, Sherman D, Li M. Soundlock: A novel user authentication scheme for vr devices using auditory-pupillary response. In: Network and Distributed System Security Symposium (NDSS). 2023