Securing Shared State in Multi-User Augmented Reality

Jiasi Chen[*]
University of Michigan, Ann Arbor

Carter Slocum[†]
University of California, Riverside

Yicheng Zhang[†]
University of California, Riverside

Erfan Shayegani[†]
University of California, Riverside

Pedram Zaree[†]
University of California, Riverside

Nael Abu-Ghazaleh[‡]
University of California, Riverside

Abstract

In multi-user AR experiences, some form of shared state is needed to save the state of the application and propagate it to the participating devices. Examples of shared state include Google ARCore's CloudAnchors, which are stored in the Google Cloud, or Mapillary's crowd-sourced street view data that could power future AR experiences. In this position paper, we highlight attacks that can disrupt the shared state and result in adverse impacts on users, such as causing workers to view incorrect AR safety signs in construction zones. We summarize possible mitigations at the data input, cloud storage, and output sides, and touch on their deployment challenges.

Index Terms: Augmented reality, multi-user, shared state, image spoofing

1 Scenario: Multiple users in a shared AR experience

The scenario under consideration is multiple users participating in a shared experience, with these users being geo-distributed across multiple locations. For example, consider a shared AR whiteboard application where users at one institution collaborate on a virtual whiteboard with users at another institution. These types of shared experiences with three or more users typically require a central data fusion point, or repository, to coordinate what is happening in application and propagate it to all the users. We call this central location the "shared state". Examples of shared state that currently exist are CloudAnchors in Google ARCore [1], which stores ephemeral visual and virtual data in the cloud, as well as Mapillary [3], which stores crowdsourced street view data that could be used to power outdoor AR experiences. The focus of this position paper is to ask the question: ***What happens if there are disruptions to the shared state in multi-user AR experiences?***

2 What happens if shared state is disrupted?

In recent work [4], different types of attacks on shared state have been demonstrated. On the left in Figure 1 is an attack on Google ARCore. What happens here is that a benign user initially writes a hologram (the colorful 3D axis) to the physical location on the floor. Subsequently, an attacker shows a picture of this physical location to the AR headset, which accepts the spoofed image and renders this hologram, despite the attacker not being physically present in the room. Although this example is simple, one can imagine the same mechanism being used by an attacker to view private holograms that are supposed to be contained to a certain physical space. Figure 2 shows an example of tampering with Mapillary's shared state. The top row shows the no attack scenario. On the bottom row, the tampering resulted in two signs, "dig safe" and "danger: underground gas line" being swapped, so that the victim sees the "dig safe" sign in the unsafe pipe location and the "danger" sign in this perfectly safe field. Such an attack could impact AR-equipped construction workers who use the virtual signage to determine safe areas to dig. Note that all the above attacks were demonstrated in private sessions or sandboxes and hence did not harm regular users.

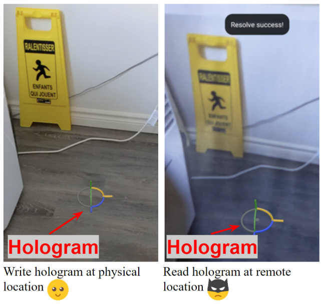
Resolve success!
Hologram
Hologram
Write hologram at physical location 😮
Read hologram at remote location

Figure 1: *Left:* A victim places a hologram in front of a yellow sign. *Right:* An attacker is able to view the hologram from a photograph without being physically near the yellow sign.

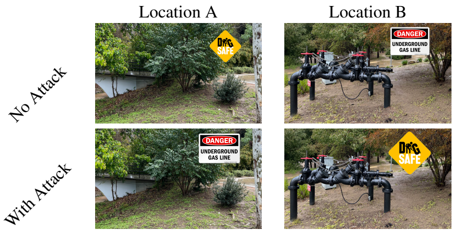

Figure 2: A "safe to dig" sign is wrongly placed next to an underground pipe.

3 Possible mitigations

There are several possible mitigations. One idea is to use detect input spoofing using multimodal sensors. For example, using the HoloLens 2 depth camera and the RGB camera, one could compare images from the respective cameras to try to determine out if input spoofing took place. However, this requires additional resources in the form of computations and sensor usage. On the output side, one could try to deploy existing techniques from the literature [2] to define policies to prevent bad outputs. The question, then, is how to detect these policy violations in the context of multi-user AR with shared state.

4 Conclusions

We look forward to a productive discussion on these and other ideas with participants at the SafeAR workshop.

[*] e-mail: jiasi@umich.edu
[†] e-mail: {csloc001,yzhan846,sshay004,pzare003}@ucr.edu
[‡] e-mail: nael@cs.ucr.edu

References

[1] Google. ARCore Cloud Anchor API. https://developers.google.com/ar/develop/cloud-anchors/management-api. 1

[2] K. Lebeck, K. Ruth, T. Kohno, and F. Roesner. Securing augmented reality output. In *IEEE Symposium on Security and Privacy (SP)*, 2017. 1

[3] Mapillary. Mapillary: make better maps. https://www.mapillary.com/. 1

[4] C. Slocum, Y. Zhang, E. Shayegani, P. Zaree, N. Abu-Ghazaleh, and J. Chen. That doesn't go there: Attacks on shared state in multi-user augmented reality applications. In *USENIX Security Symposium*, 2024. 1