Preserving Privacy in Human-Machine and Agent Interaction

Kevin Butler

UF UNIVERSITY of FLORIDA
Duke UNIVERSITY
TEXAS The University of Texas at Austin
UC SANTA CRUZ

- Topics:
  - Privacy-preserving eye-tracking (IEEE TVCG/VR’21)
  - Privacy for ROS (early work in progress)

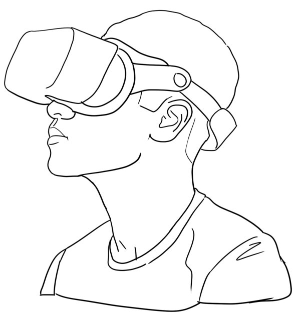
- Collaborators (UF unless otherwise noted):
  - Brendan David-John, Diane Hofstedt (Mozilla), Eakta Jain, Washington Garcia, Aaditya Prakash

Human-Machine Interaction

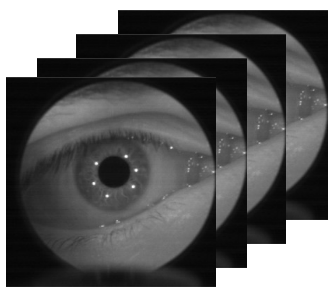
- Current state of deployed UASs currently involve significant human interaction (<=L3 autonomy)
- Autonomous systems will potentially learn from simulation data informed by human interaction
- Augmented reality (AR) systems can assist near-term operations while virtual reality (VR) simulators are standard for training
- What risks to privacy are incurred in these systems?

Eye Tracking Overview

User X



Gaze Data Stream
<x, y, t>



Eye Tracker

Applications

Aggregate-level: Area-of-interest Analysis

Event-level: Redirected Walking

Sample-level: Foveated Rendering

Sample-level: Biometric Identity

Event-level: Health Conditions

Aggregate-level: Sexual Orientation

UF UNIVERSITY of FLORIDA

Duke UNIVERSITY

TEXAS The University of Texas at Austin

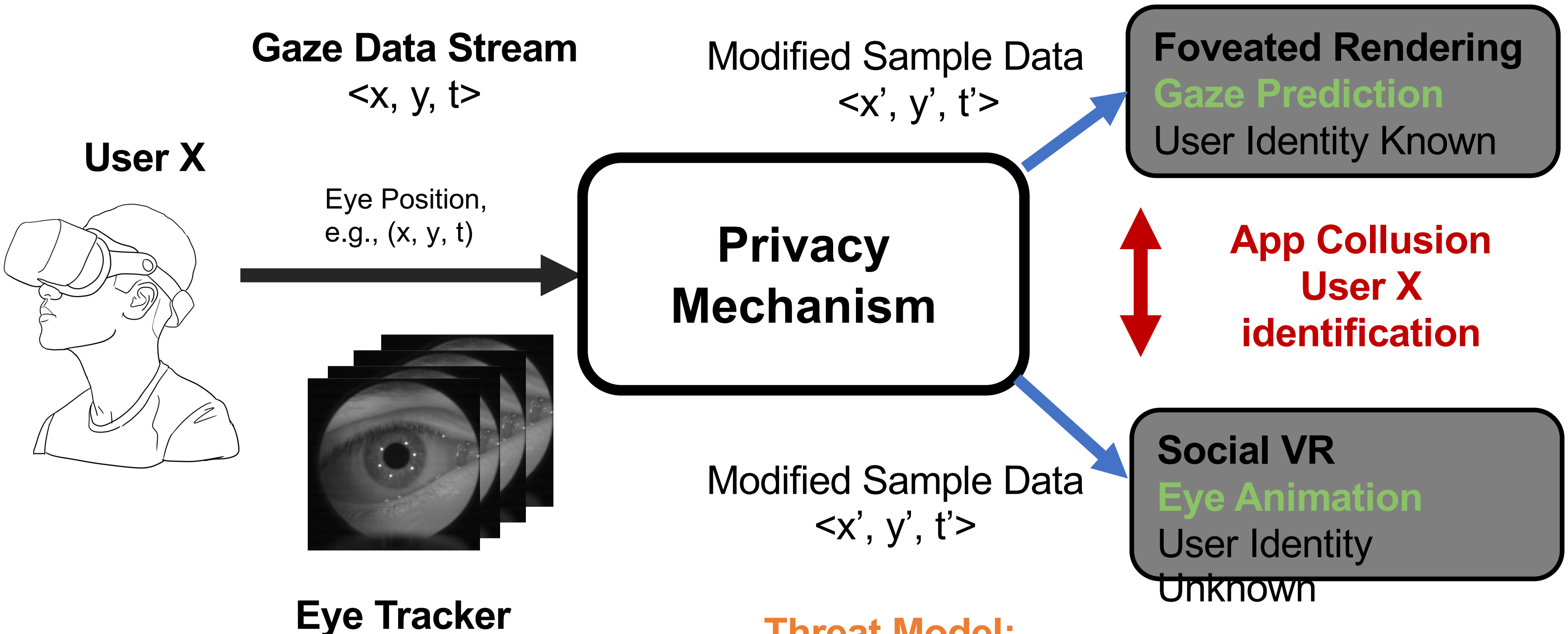
UC SANTA CRUZ

Contribution

- Establish a framework for streaming eye-tracking data to support mixed reality applications while mitigating risks given a threat model.

- Scenario 1: Applications that require aggregate/event-level data
  - Threat: Sample-level data
  - Solution: Gatekeeper API
- Scenario 2: Applications that require sample-level data
  - Threat: Biometric Identification by Colluding Apps
  - Solution: Privacy Mechanisms

Privacy Mechanism



Threat Model:
Trusted Platform, Untrusted Applications

Privacy Mechanism Definitions

- Consider privacy mechanism data as a time series, tuple comprises gaze positions (x,y) and event label for G gaze position $e: X = \{(x_1, y_1, t_1, e_1), ...(x_G, y_G, t_G, e_G)\}$
- Additive Gaussian noise: independently sample from Gaussian distribution for horizontal and vertical gaze position

$$X' = \{(x_1 + N(0, \sigma), y_1 + N(0, \sigma), t_1, e_1), ..., (x_G + N(0, \sigma), y_G + N(0, \sigma), t_G, e_G)\}$$

- Temporal downsampling: reduce temporal resolution of eye-tracking data stream by reducing sampling rate by scaling parameter *k*: $X' = \{(x_{(K \cdot p)+1}, y_{(K \cdot p)+1}, t_{(K \cdot p)+1}, e_{(K \cdot p)+1}), ...\}$
- Spatial downsampling: divide scene into equirectangular domain and parameterize into steps s.t. gaze positions compute as $(\lfloor \frac{x}{\delta_x} \rfloor \cdot \delta_x, \lfloor \frac{y}{\delta_y} \rfloor \cdot \delta_y, t)$

  where $\delta_y = \frac{180}{M}, \delta_x = \frac{360}{N}$

Evaluation

New Dataset

Datasets:

- ET-DK2 (360◦ Images)
  - 19 Individuals, 50 Stimuli
- **VR-Saliency** (360◦ Images)
  - 130 Individuals, 8 Stimuli
- **VR-EyeTracking** (360◦ Images)
  - 43 Individuals, 148 Stimuli
- **360_em** (360◦ Images)
  - 13 Individuals, 14 Stimuli
- **DGaze** (3D Rendered Scenes)
  - 43 Individuals, 2 Stimuli

Mechanisms:

- Additive Gaussian Noise
- Temporal Downsample
- Spatial Downsample

Metric:

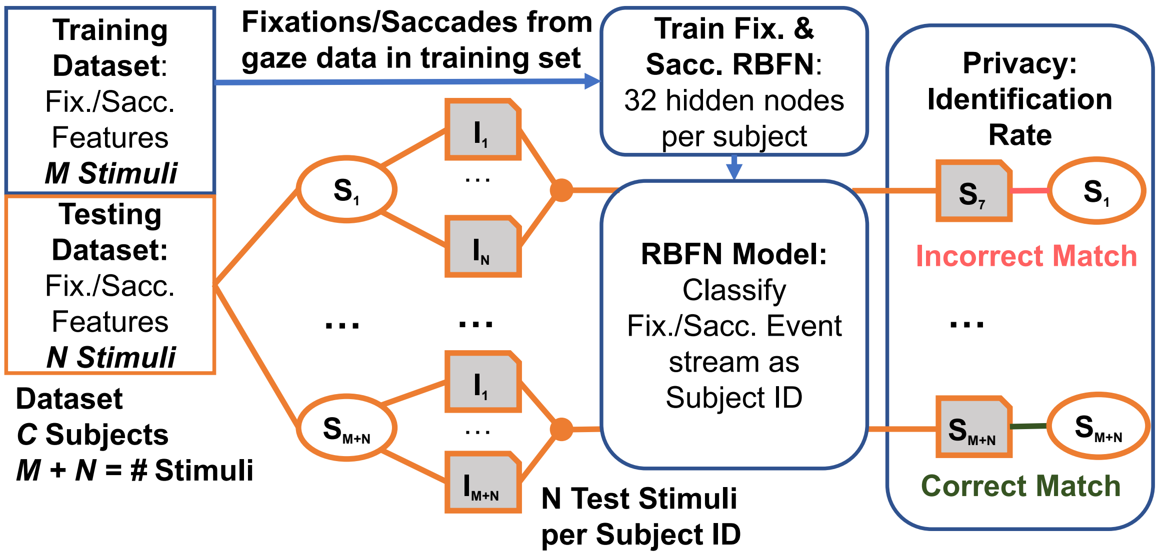
Identification Rate

$$\frac{\#\ Stimuli\ Correctly\ Predicted}{Total\ \#\ of\ Stimuli\ in\ Test\ Set}$$

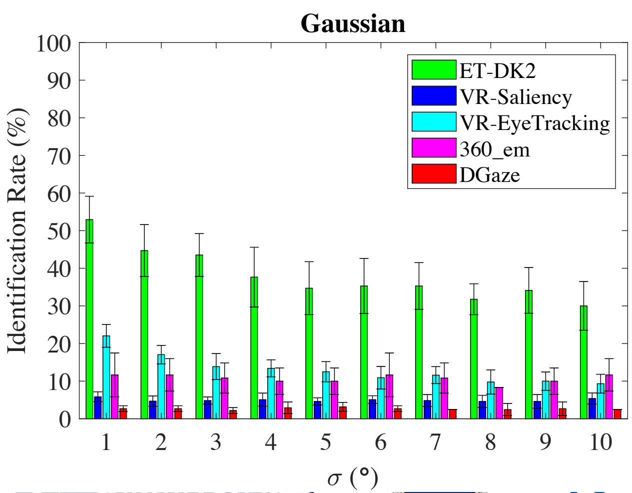
Biometric Re-Identification Classifiers

- Radial Basis Function networks for classifying fixation and saccade events
  - Analogous to neural network with input layer representing feature vector $\vec{x} \in \mathbb{R}^p$ with p fixations/saccade features , activation function $\phi_i(\vec{x})$ and real-valued activation weights $w_{i,c}$, where $i \in [1, 2, \ldots, m]$ and $j \in [1, 2, \ldots, C]$
  - Class similarity score $Score_c(\vec{x}) = \sum_{i=1}^{m} w_{i,c} \cdot \phi_i(\vec{x})$
  - Hidden node activation function $\phi_i(\vec{x}) = e^{-\beta_i ||\vec{x} - \vec{\mu}_i||^2}$
  - K-means clustering on training set to determine representative feature vectors
  - Activation weights trained by using Moore-Penrose inverse when setting up network as linear system (gradient descent also possible)

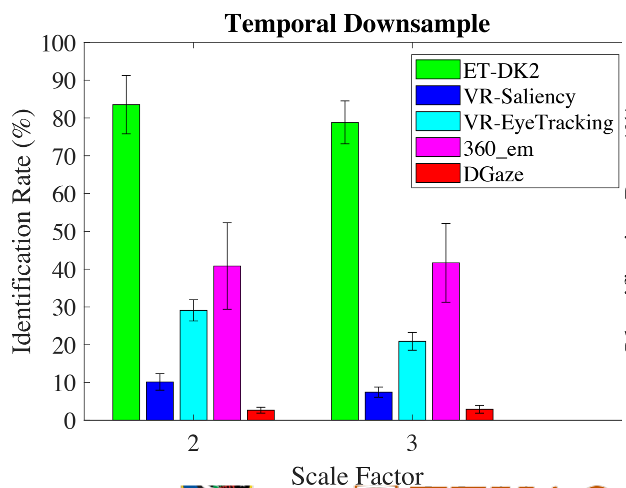
Results



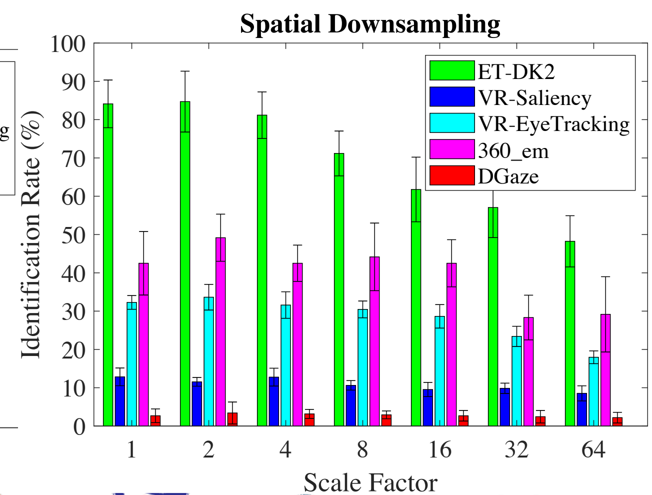
Training Dataset: Fix./Sacc. Features M Stimuli
Fixations/Saccades from gaze data in training set
Train Fix. & Sacc. RBFN: 32 hidden nodes per subject
Privacy: Identification Rate
Testing Dataset: Fix./Sacc. Features N Stimuli
S1
I1
IN
S7
S1
Incorrect Match
RBFN Model: Classify Fix./Sacc. Event stream as Subject ID
Dataset C Subjects M + N = # Stimuli
SM+N
I1
IM+N
SM+N
SM+N
Correct Match
N Test Stimuli per Subject ID



Gaussian
Identification Rate (%)
σ (°)
ET-DK2
VR-Saliency
VR-EyeTracking
360_em
DGaze
0
10
20
30
40
50
60
70
80
90
100
1
2
3
4
5
6
7
8
9
10

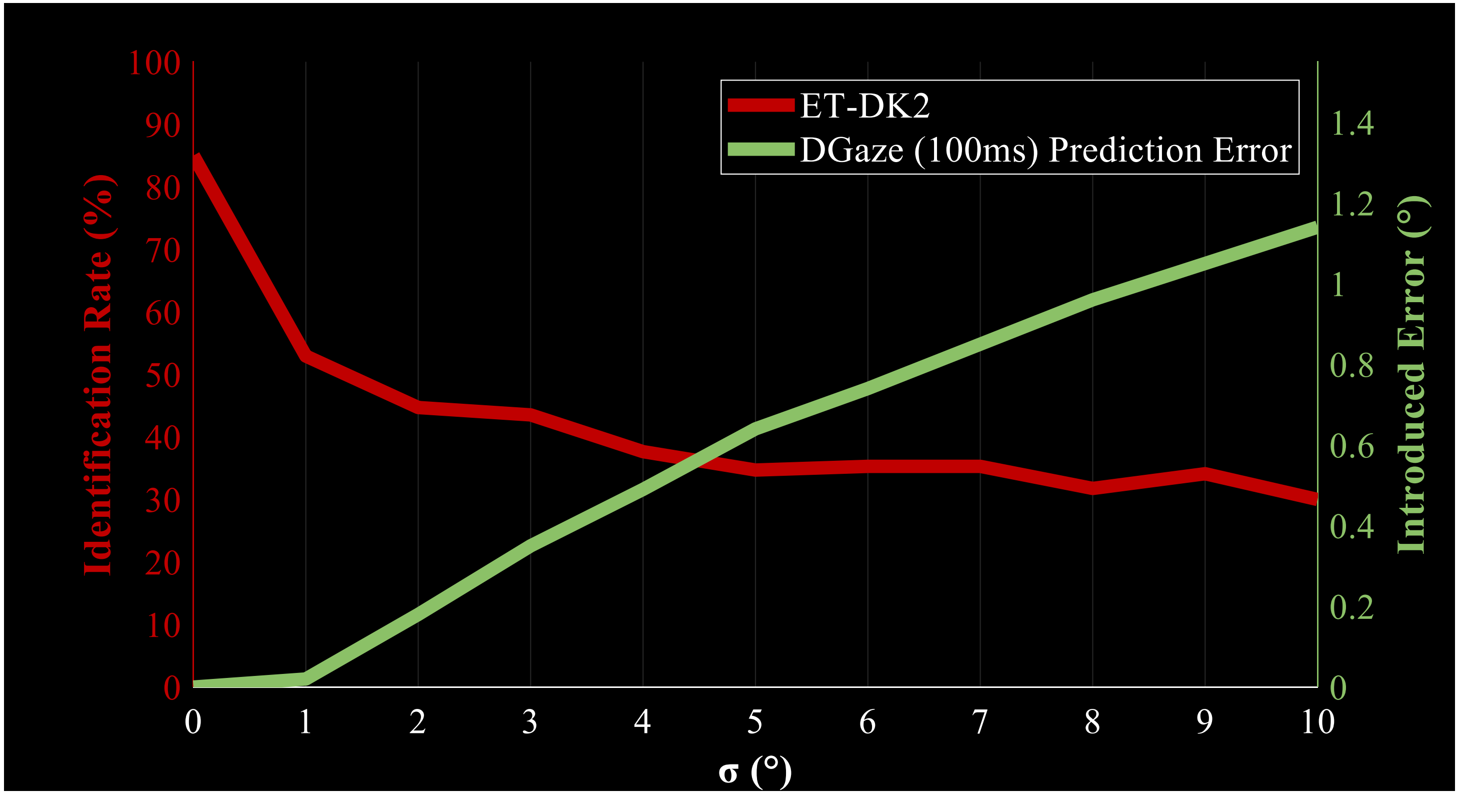


Temporal Downsample
Identification Rate (%)
Scale Factor
ET-DK2
VR-Saliency
VR-EyeTracking
360_em
DGaze
0
10
20
30
40
50
60
70
80
90
100
2
3



Spatial Downsampling
Identification Rate (%)
Scale Factor
ET-DK2
VR-Saliency
VR-EyeTracking
360_em
DGaze
0
10
20
30
40
50
60
70
80
90
100
1
2
4
8
16
32
64

Privacy-Utility Trade-off: Gaze Prediction



UF UNIVERSITY of FLORIDA Duke UNIVERSITY TEXAS The University of Texas at Austin UC SANTA CRUZ

Summary

- A Gatekeeper model can be used to protect biometric features in gaze data streams for many applications.
- Standalone privacy mechanisms can be applied for applications when sample-level data is needed.
- **Takeaway:** Privacy-preserving frameworks are necessary to address threats to privacy from behavioral data in mixed reality.

Privacy-Preserving Computation

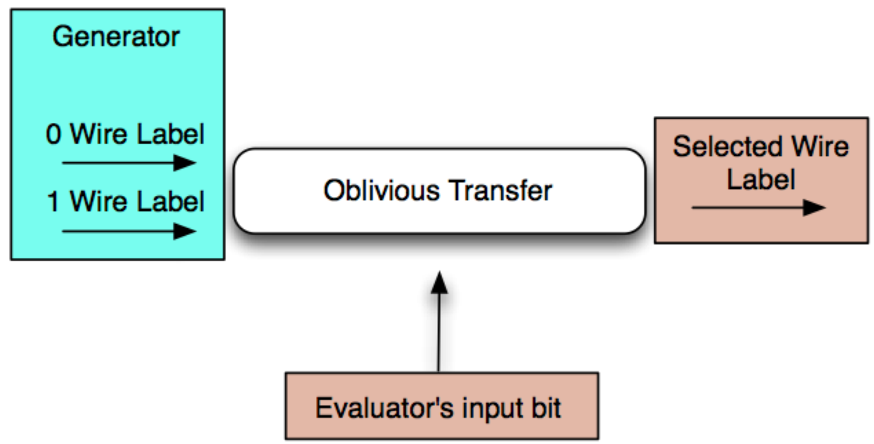
- How to ensure that data communicated by autonomous agents stays secure and private?
- Communication privacy:
  - TLS
  - Differential privacy
  - Private Information Retrieval
  - ORAM
- Computation privacy:
  - TEEs
  - FHE
  - SMC
    - Two-party (garbled circuits)
    - >2 parties (shared secrets)

Secure Multiparty Computation

- Allow joint computation of a function without revealing input from either party
- Cryptographically secured through the use of *garbled* Boolean circuits and *oblivious transfer* of data from circuit generator to evaluator

Garbled Gate
Lock($A_0 \| \| B_0$, O_0)
Lock($A_0 \| \| B_1$, O_0)
Lock($A_1 \| \| B_0$, O_0)
Lock($A_1 \| \| B_1$, O_1)

lock Output O under inputs A and B

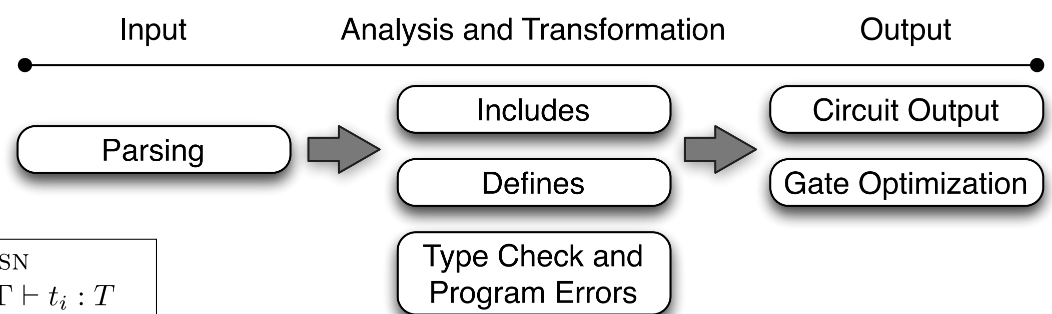


Circuit Evaluation

- Generator sends evaluator the input wire keys
- 1-of-2 oblivious transfer for each input wire
  $k_0 = (v - x_0)^d \bmod N, k_1 = (v - x_1)^d \bmod N$
- Evaluator decrypts output gates $E_{k_{r,*}}(E_{k_{l,*}}(k_{v_{bit_l,bit_r}}))$
  - $k_{l,*}$ and $k_{r,*}$ are keys the evaluator has
  - $k_{v_{bit_l,bit_r}}$ is the garbled truth table entry selected by the point and permute bits bit_l and bit_r

Frigate Compiler

- Validated garbled circuit compiler, along with the Battleship semi-honest interpreter environment
- Used standard compiler practices
  - Validation testing
  - Proper data structures, e.g., AST
  - Created a formal description of how operations should function



ADD
$$\frac{\Gamma \vdash t_i : Num_{L_i}}{\Gamma \vdash t_1 + t_2 : Num_{L_i}}$$

LESS
$$\frac{\Gamma \vdash t_i : Num_{L_i}}{\Gamma \vdash t_1 < t_2 : Num_1}$$

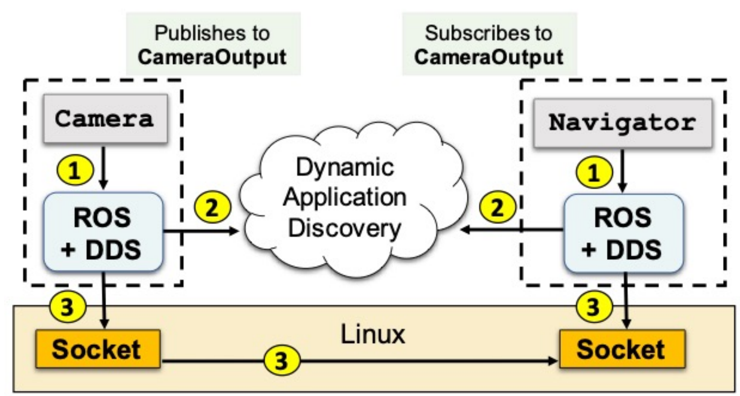
ASSN
$$\frac{\Gamma \vdash t_i : T}{\Gamma \vdash t_1 = t_2 : T}$$

IF-ELSE
$$\frac{\Gamma \vdash t_i : T \quad \sigma : Num_1}{\Gamma \vdash \ \texttt{if}\ (\sigma)\{t_1\}\ \texttt{else}\ \{t_2\} : T}$$

FUNC-CALL
$$\frac{\Gamma \vdash t_i : T_i \quad f : F}{\Gamma \vdash f(t_0 ... t_{n-1}) : R}$$

Preserving Privacy in ROS

- Little in the way of existing solutions for privacy-preserving computing on ROS
- Privaros (Beck et al., CCS'20)
  - Privacy-preserving policy enforcement for drones
  - Relies on system security mechanisms: mandatory access controls on OS for inter-app communication and TrustZone TEE for attesting the SW stack
  - Computation of data not performed in oblivious fashion

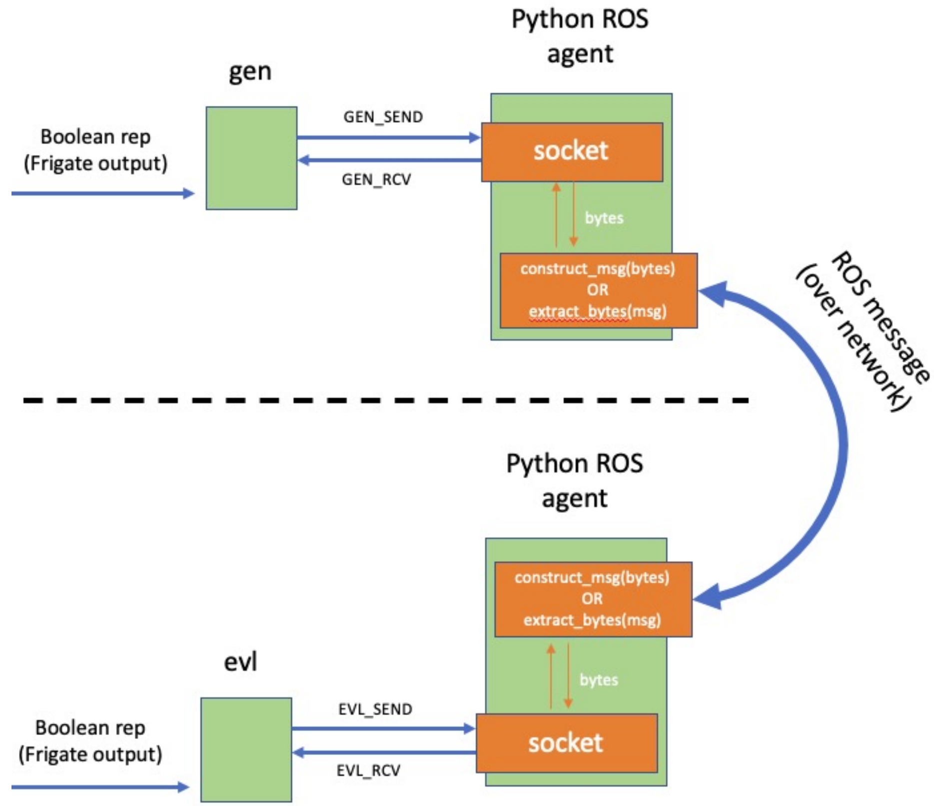


① Every application on ROS links against the library. The dotted lines show the process boundary. An application registers its topics via the ROS library; ② A decentralized protocol discovers and identifies applications with matching topics; ③ ROS sets up socket communication via the underlying OS for the applications.

ROS publish/subscribe model (Beck et al.)

Approach: SMC on ROS

- Use Frigate compiler to generate circuits and Battleship interpreter (currently considering the semi-honest adversary model)
- Developed Python-based ROS agent for converting GC data into pub/sub data model suitable for ROS messages
- Early feasibility results currently underway



Next steps

- Security analysis of oblivious transfer in ROS
- Increase complexity of garbled circuit evaluation (e.g., privacy-preserving direction finding)
- Update efficiency of GC mechanisms and consider n-party sharing
- Dive into FHE (incoming student)
- ROS Fuzzing
  - Examine new techniques from the fuzzing community and whether they provide a new assessment of ROS2 security