Chronos: Timing Interference as a New Attack Vector on Autonomous Cyber-physical Systems

Ao Li
Washington University in St. Louis
ao@wustl.edu

Jinwen Wang
Washington University in St. Louis
jinwen.wang@wustl.edu

Ning Zhang
Washington University in St. Louis
zhang.ning@wustl.edu

ABSTRACT

Timing property plays a vital role in the Cyber-Physical System(CPS) due to its interaction with the physical world. The smooth operation of these robotic systems often relies on an accurate and timely perception and actuation of the physical world.

In this poster, we demonstrated a unique new class of attack, Chronos, that exploits timing interference to cause system destabilization in cyber-physical systems. Using a compromised non-privileged non-critical task on the system, we launch timing interference attacks on both drone and autonomous vehicle platforms. Through both open-loop and close-loop testing on the end-to-end stack, we showed that the timing attack could lead to complete loss of control of the autonomous system, crashing them onto the surroundings when there is no software vulnerability.

To further understand this novel attack vector, we perform preliminary investigations on the localization component of these two platforms, because they both make use of well-known simultaneous localization and mapping (SLAM) algorithms that depend on timing-sensitive multimodal data from different sensors. Building on the insights from the case study, we present our formulation of the timing attack surface and highlight future directions.

CCS CONCEPTS

• **Security and privacy → Software security engineering**.

KEYWORDS

Timing attack; Timing sensitivity analysis; Simultaneous Localization and Mapping; Robotics security

ACM Reference Format:
Ao Li, Jinwen Wang, and Ning Zhang. 2021. Chronos: Timing Interference as a New Attack Vector on Autonomous Cyber-physical Systems. In *Proceedings of Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3460120.3485350

A demo video is available at https://youtu.be/7oyXs1yW2xw. This work was supported in part by US National Science Foundation under grants CNS-1837519 and CNS-2038995.

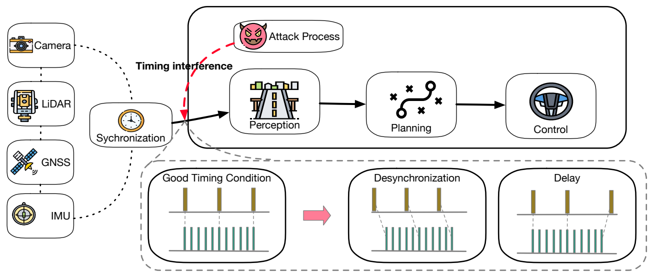


Figure 1: The overview of the proposed attack.

1 INTRODUCTION

CPS plays an increasingly important role in our society, making profound changes to our daily life via innovations in self-driving cars, medical robots, smart factories, and many other domains. A fundamental difference between CPSs and conventional IT systems is their interactions with the physical world. While many computation tasks in IT systems can be suspended for an extended period of time, calculations in CPSs often have a strong time affinity, since time continues to elapse in the physical world. Computationally correct, but untimely results often have little to no value for system control, and can even destabilize the physical system.

Existing works in cyber-physical timing security generally have two types of assumptions. The first type is adapted by Mahfouzi et. al. in the bufferfly attack [14], where it assumes the existence of software feature (vulnerability) that can be triggered using malicious input to significantly change the timing of system actuation. In [14], this vulnerability was exploited to trigger the control jitter condition causing control system destabilization. The second type is adapted by Luo et. al. in [13], where the attacker can compromise a non-privileged and non-critical task in the autonomous CPS to launch side-channel attack to infer the travel trajectory.

In this paper, we make the assumption a non-privileged non-critical task can be compromised by the attacker, however, the control system software is bug-free. The key question we ask is whether it is feasible to exploit timing interference as an attack vector to cause control errors in cyber-physical systems. To the best of our knowledge, we are the first to demonstrate and dissect the software timing interference as an effective attack vector on the real-world cyber-physical system stack.

As shown in Figure 1, the attacker who controls a non-privileged task can interfere with the timing of the system by executing performance interfering tasks, such as random memory access to pollute the cache, to degrade the system control performance, or even completely disabling the autonomous system as shown in our demo video. In order to dissect the attack and to gain an in-depth understanding of this new attack surface, we choose to conduct a case study on perception subsystem of the CPS. In both the drone [1] and autonomous vehicle platform, Jackal UGV [4], the perception module makes use of the SLAM for its localization. Furthermore, SLAM is a representative robotic algorithm, which involves filter-based estimation, multiple sensor fusion. Since SLAM is one of the first modules in the control pipeline, the precision of its output will directly impact the system performance. Lastly, SLAM has been widely applied in various domains such as AR devices, drones, unmanned guided vehicles, self-driving cars, or humanoid-robot. We believe the insights we develop by studying algorithms in this component can be applied to other domains. We found that there are two factors enabling the attack.

1) Desynchronization in Sensor Fusion: The first type of error comes from the desynchronization of different sensor inputs. Time synchronization is critical in sophisticated robotic systems such as self-driving cars. For example, many perception tasks, such as object detection, increasingly fuse 3D information from LiDARs and 2D information from cameras. If the LiDAR and the camera are well-synchronized, the corresponding 3D bounding box detected from LiDAR data and the 2D bounding box detected from camera data are well-matched. Hence, manufacturers leverage hardware to support inter-sensors synchronization. For example, Nvidia Xavier AGX [6] triggers the LiDAR and camera at the same time. Apollo [2] recommends the synchronization between LiDAR [7] and GNSS [5] using Pulse-Per-Second. However, we found that even when the sensing is perfectly synchronized in hardware, it is still possible to use a timing attack to interfere with the timing of sensor polling for different sensors. This is possible because there is an implicit assumption on the software execution timing when the program is written, and Chronos undermines this assumption.

2) Delay in Control: As shown in Figure 1, delay in computation task for control can impact the accuracy of robotic system perception, which is basis for next control loop actions. An example is that a delay in pose estimation can degrade the quality of the most recently constructed map, leading inaccurate key frame recognition. Another example is that residuals caused by IMU data increase significantly over time, when the task in charge of correcting the residuals is delayed, the final result continuously deviates from the ground truth. Furthermore, even though these errors from timing violation may be negligible for a single frame, they accumulate over time due to the unique continuous nature of many state estimators.

We evaluated the attack on four well-known open-source SLAM systems, we found that the actual robotic system is not as stable as the assumption. It is possible for robots to lost their location and finally crash due to the violation of timing properties.

2 TIMING VULNERABILITIES IN SLAM

To develop a better understanding on the source of the errors due to timing inference, we study the architecture (shown in Figure 2, the component with circle or triangle means it suffers from the delay or desynchronization respectively.) and implementation details of SLAM system, found four categories of errors observed in our attack.

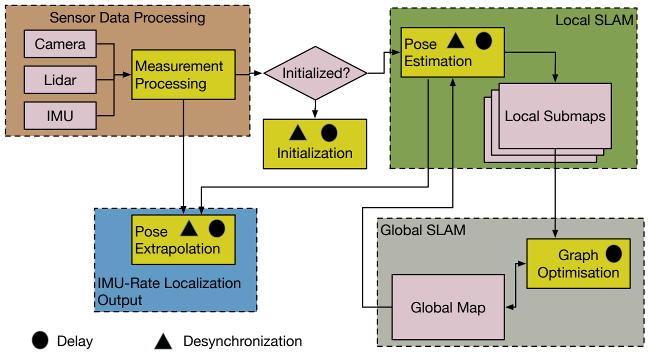


Figure 2: The overview of a VIO/SLAM system. Each yellow block represents a building block of SLAM.

Pose Estimation Error. Modern SLAM usually models the matching between adjacent frames via Maximum a posteriori estimation(MAP). The accuracy of the solver relies on the quality of the initial guess given by inertial information.
Pose Extrapolation Error. Inertial data is forward propagated to achieve a higher localization rate to meet the requirement of the controller. An execution time increase in the visual processing subsystem causes inaccuracy of the fused visual-inertial state estimation [17], since the IMU bias cannot be corrected in time.
Initialization Error. Most SLAM systems require an initialization step to calibrate the bias of IMU. General monocular SLAM estimates the depth of landmarks at initialization. Wrong initialization may affect negatively the whole trajectory or cause re-initialization.
Graph Optimization Error. Modern SLAM often uses the formalism of factor graphs to reason about the interdependence among each key submap. It is also formulated as a MAP problem so that it is vulnerable to the same problem as pose estimation.

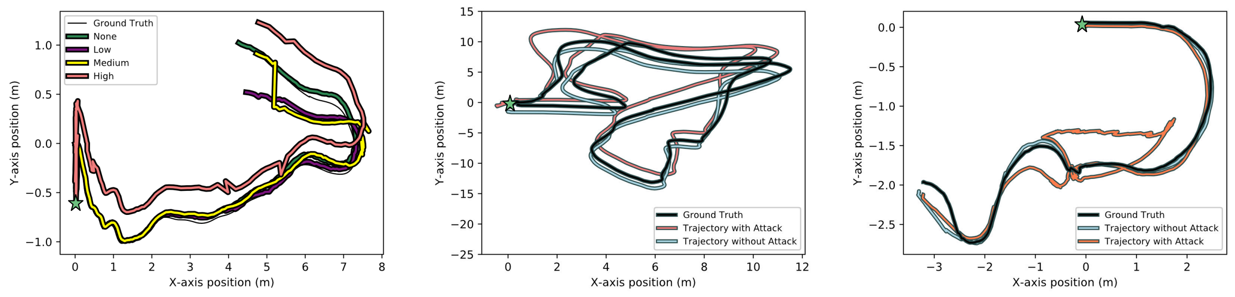
3 ATTACK EVALUATION

3.1 Experimental Setup

Our evaluation is conducted in two different setups. An open-loop evaluation with the EuRoC MAV dataset [8] aims to analyze the timing sensitivity of each error source. And a closed-loop evaluation with simulated Jacal UGV and office scenarios in Gazebo [12]. We addressed our attack on four targeted SLAM system which are two image-based SLAM systems: VINS-Mono [16] and ORB-SLAM [15] and two LiDAR-based SLAM systems: Cartographer [3, 10] and Adaptive Monte-Carlo Localization [9]. The entire software stack was performed on a laptop with Intel i7-7700HQ CPU and 8GB RAM. The simulation is running on a host PC with Nvidia GTX 1060 GPU. A hardware-in-the-loop test environment is implemented via Ethernet ports.

3.2 Attack Prototype

To interfere with the execution time of victim tasks, we implement a proof of concept adversary task [11] that writes an array with cache line size jump in each iteration in an infinite loop. With the adversary task, the cache used by victim tasks would immediately be evicted due to the contention. We launch the adversary tasks



(a) Trajectories deviation at different attack levels (b) Open-loop evaluation of attack on drone (c) Closed-loop evaluation of attack on vehicle

Figure 3: Experimental results. The start points of trajectories are marked with green star.

Table 1: Timing anomalies under different attacks

	Exe.Time	*RPE*	*ATE*	*Max.ATE*
None	1*X*	0.0075	0.098	0.348
Low	1.5*X*	0.0141	0.192	0.913
Medium	2.0*X*	0.0219	0.359	2.088
High	2.5*X*	0.0262	0.366	3.252

RTE: Relative Pose Error
ATE: Absolute Trajectory Error

with default scheduling priority. The number of attack tasks varies from 0 to 6, which indicates the intensity of the attack.

3.3 Timing Sensitivity Analysis

To demonstrate SLAM systems are prone to produce timing anomalies due to resource contention, we configured several system overheads with different contention intensities. The four SLAM systems are tested with all sequence EuRoc dataset [8]. The result in Table 1 shows that the execution time of SLAM could be doubled under the exhaustion of resources. As to absolute error, the beginning sequences of these trajectories are illustrated in Figure 3(a). The absolute trajectory error is accumulated by the relative errors during the travel. The localization turns useless since the estimated trajectory deviates more than 0.3 meters from the ground truth.

3.4 Attack Effectiveness

To evaluate the effectiveness of the attack, we conduct both open-loop and closed-loop evaluations. Open-loop testing relays recorded data and evaluates the control deviation over individual small time periods, and therefore highlights the differences over the control output. Our experiment on VINS-Mono[16] is shown in Figure 3(b), the attack causes the system to deviate significantly from the ground truth with a maximum of absolute trajectory error greater than 1 meter. In practice, this will lead to drone crashing.

We perform closed-loop evaluation, on a Jackal UGV running Cartographer for localization. The difference of the tracked trajectory for the system with and without the attack is shown in Figure 3(c). The vehicle without attack tracks closely the reference path, while the red trajectory under attack lost its localization and crashed to the wall.

4 CONCLUSION AND FUTURE WORK

In this paper, we presented our preliminary investigation on leveraging timing interference as a new novel attack vector. Our evaluation shows that despite the fact that the proof-of-concept attack implementation is naive, the attack turns out to be highly effective on both autonomous vehicles and drones. Using SLAM as a target subsystem in CPS to understand the timing attack, we analyzed and categorized four types of errors.

Given the growing importance of autonomous cyber-physical systems in our society, it is important to further study, characterize, model, and mitigate this new attack vector, which works out of the box without any existing software vulnerabilities that our current defenses are developed to handle. In the future, we aim to develop a further characterization of the attack by considering more advanced attack models on additional platforms and to develop techniques to mitigate such risks on cyber-physical systems.

REFERENCES

[1] Asctec firefly mav. http://www.asctec.de.
[2] Baidu apollo open-source self-driving project.
[3] Cartographer. https://google-cartographer.readthedocs.io.
[4] Jackal ugv. https://clearpathrobotics.com.
[5] Novatel gnss inertial navigation systems.
[6] Nvidia xavier agx. https://www.nvidia.com.
[7] Velodyne hdl-32e. https://velodynelidar.com/products/hdl-32e/. Accessed: 08-15.
[8] Michael Burri et al. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research.*
[9] Dieter Fox. Kld-sampling: Adaptive particle filters and mobile robot localization. *NIPS*, 2001.
[10] Wolfgang Hess et al. Real-time loop closure in 2d lidar slam. In *ICRA*. IEEE, 2016.
[11] Dan Iorga et al. Slow and steady: Measuring and tuning multicore interference. In *RTAS*. IEEE, 2020.
[12] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IROS*. IEEE, 2004.
[13] Mulong Luo et al. Stealthy tracking of autonomous vehicles with cache side channels. In *USENIX Security*, 2020.
[14] Rouhollah Mahfouzi et al. Butterfly attack: Adversarial manipulation of temporal properties of cyber-physical systems. In *RTSS*, 2019.
[15] Raul Mur-Artal et al. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics.*
[16] Tong Qin et al. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics.*
[17] Yipu Zhao et al. Closed-loop benchmarking of stereo visual-inertial slam systems: Understanding the impact of drift and latency on tracking accuracy. In *ICRA*. IEEE, 2020.