Reviewer Comments:

Reviewer: 1

Recommendation: AQ - Publish With Minor, Required Changes

Comments:

Thank you very much for the updated version of the document.

== Clarifications ==

Abstract: line 16 could give the impression to readers that other hardware manufacturers also provide SGX: "hardware manufacturers such as Intel released a new processor feature called Software Guard eXtension (SGX)".

I would suggest to make it clear that hardware vendors propose TEEs; Intel has implemented one called SGX.

**Author response:** Thanks for your kindly reminder. We reclaim the statement: “hardware vendors propose Trusted Execution Environment (TEE). Particularly, Intel has released a new processor feature called Software Guard eXtension (SGX)”.

Threat model:

-I would suggest to explicitly state that the attacker does not have access to the hardware. This would ease readability.

**Author response:** Yes. This is align with our assumption that the physical attacks are out of scope: “The adversary cannot access the physical machine and modify its hardware setup.”

-page 7: "Such a sibling thread will block the possibility of side-channel attacks when hyper-threading is active.". Hyperthreading should be disabled for security reasons (see https://marc.info/?l=openbsd-tech&m=153504937925732&w=2for example).

**Author response:** Thanks for the information. Yes, the Hyperthreading/SMT should be disabled for security reasons. The authors of Hyperspace[1] proposed using a sibling thread to occupy the slot of SMT, thus preventing side-channel attacks when SMT is not disabled. We followed their claim. In the updated version, we cite the work for better understanding.

[1] Racing in Hyperspace: Closing Hyper-Threading Side Channels on SGX with Contrived Data Races. IEEE Symposium on Security and Privacy 2018: 178-194

Trusted time: Intel SGX v2 has access to trusted time (RDTSC), see section "39.6.1 Illegal Instructions" of the Intel® 64 and IA-32 Architectures Software Developer’s Manual. How does it affect your solution?

**Author response:** Although SGX v2 provides the ability to access RDTSC for enclaves, the malicious kernel can still change the TSC value to make it untrusted. Aurora provides an attack-detection method which can also benefit for SGX v2.

It would be interesting to see more details about Intel PVAP. Can it be integrated with Aurora? Can it provide additional guarantees?

**Author response:** Intel PAVP is orthogonal to Aurora, as PAVP leverages -3 privilege which is Intel Management Engine， whereas Aurora leverages -2 SMM mode. When using PAVP, there is no need to use Aurora for an additional secure channel.

== Evaluation ==

Table 4 (serial port performance), what is the performance with a larger payload (e.g. 1MB)?

**Author response:** The performance of serial port for transferring 1MB (1024KB) is 3.95s (Linux) v.s. 3.43s (Aurora), with 13.2% speedup. The results are shown in Table IV.

TPC-H results, why are some requests not working?

**Author response:** It depends on the soundness of SQLite porting. Porting SQLite to fully work in SGX enclave is not our focus in this work. We run TPC-H tests and show completed query results in Table VII.

== Minor comments ==

-Will the authors make the code available? It would be very beneficial for the community for building new secure systems with trusted I/O.

**Author response:** The code will be made available on Github: https://github.com/Maxul/Aurora.

We will make comprehensive documentation for future researchers to build new security systems upon Aurora.

-introduction: rely \*on\* the clock value from untrusted systems

-p6, afterward\*s\*

**Author response:** Thank you, they are fixed.

Additional Questions:

1. Is the topic appropriate for publication in these transactions?: Adequate Match

1. Is the paper technically sound?: Yes

2. How would you rate the technical novelty of the paper?: Novel Enough for Publication

Explain: The problem of trusted I/O paths is an important problem for secure computing. The authors present an interesting and novel approach that combines Intel SGX with SMM. I believe the approach is worth being published.

3. Is the contribution significant?: Incremental

4. Is the coverage of the topic sufficiently comprehensive and balanced?: Yes

5. Rate the Bibliography: Satisfactory

1. How would you rate the overall organization of the paper?: Satisfactory

2. Are the title and abstract satisfactory?: Yes

3. Is the length of the paper appropriate? If not, recommend what should be added or eliminated.: Yes

4. Are symbols, terms, and concepts adequately defined?: Yes

5. How do you rate the English usage?: Satisfactory

Reviewer: 2

Recommendation: AQ - Publish With Minor, Required Changes

Comments:

Thank you for revising the draft. Most of my major comments seem to be addressed in the updated version. The paper needs some smaller edits, as listed in my review.

Additional Questions:

1. Is the topic appropriate for publication in these transactions?: Adequate Match

1. Is the paper technically sound?: Yes

2. How would you rate the technical novelty of the paper?: Novel Enough for Publication

Explain: The paper presents Aurora architecture for trusted I/O paths for enclave applications on Intel platforms with SGX and SMM support. Specifically, the paper outlines design for securing communication with a keyboard, printer clocks, and storage in the presence of an untrusted OS. By default, Intel SGX does not guarantee secure interface to devices, and Aurora bridges this gap for applications where it is essential to have a trusted path to one or more I/O devices. The authors make use of SMM which is protected from the OS to communicate with these devices over a trusted channel.

3. Is the contribution significant?: Incremental

4. Is the coverage of the topic sufficiently comprehensive and balanced?: Yes

5. Rate the Bibliography: Satisfactory

1. How would you rate the overall organization of the paper?: Could be improved

2. Are the title and abstract satisfactory?: Yes

3. Is the length of the paper appropriate? If not, recommend what should be added or eliminated.: Yes

4. Are symbols, terms, and concepts adequately defined?: Not always

5. How do you rate the English usage?: Needs improvement

\* Paper summary

- The paper presents Aurora architecture for trusted I/O paths for enclave applications on Intel platforms with SGX and SMM support. Specifically, the paper outlines design for securing communication with a keyboard, printer clocks, and storage in the presence of an untrusted OS. By default, Intel SGX does not guarantee secure interface to devices, and Aurora bridges this gap for applications where it is essential to have a trusted path to one or more I/O devices. The authors make use of SMM which is protected from the OS to communicate with these devices over a trusted channel.

\* Comments on the revision effort

- Thank you for revising the draft. Most of my major comments seem to be addressed in the updated version. Some of them need more clarifications as listed below.

\* Following is my feedback on the revised version of the paper draft:

- Missing citations for important points in the following sections:

+ time deception attacks (Section 1)

+ monotonic counters and NVRAM (Section 1)

+ Local attestation (Section 2A)

+ SMM mode (Section 2B)

+ swell the TCB and attack vector (Section 2C)

+ existing software stacks in a secure manner (Section 2C)

+ smaller TCB is better (Section 5B)

+ Fix name for Jiang [48] to Jang (Section 8C)

**Author response:** Thanks. All citations are done.

- The two types of I/O paths are not defined but referred to in the introduction and Section 2 first para.

**Author response:** We explicitly name them in the Contributions.

- Clarify the security properties of the SMM enclave (Section 2B)?

**Author response:** In the updated manuscript, we reclaimed, “We deem SMRAM as a special enclave, because SMRAM cannot be accessed by OS/HV after initialization, and therefore protects the conﬁdentiality and integrity of SMI handler.”

- Before explaining the details of 2 categories of I/O, consider adding sentences which explain the main difference between them (Section 2C)

**Author response:** We add this after the definition (Section 2C): “The main difference between these two types is that dataprovider/-consumer path is either the sink or the source of data and must handle data in plaintext, which requires careful isolation or conﬁdential protection; Whereas the data-storage/-transmitter type acts as a medium for data at rest or in transition. It focuses on the freshness and integrity of data.”

- Section 2C, should it be trust-but-verify or verify-then-use instead of distrust-but-verify?

**Author response:** Yes, “trust-but-verify” is the correct term. We would use “verify-then-use” because it is a proper term in our paper. Thank you so much for correcting us!

- Section 2D

+ Does the UEFI driver have a standard security specification that Aurora verifies, if not, how did you decide what to check?

**Author response:** UEFI is a unified EFI/firmware standard. It has specifications for different types of devices. The current implementation of Aurora prototype checks their parameters and return values according to their internal state machines (which can be found from device manual or driver documentations/comments), in addition to runtime memory address sanitizing, as described in Section 3B.

+ The attacker will learn the I/O size and request-response sizes. Should that be explicitly scoped out of the attacker model, or does Aurora prevent such leakage?

**Author response:** Although Aurora automatically pads the message to be 4KB aligned, it still leaks information of message number. We scope them out in the assumption: “information leakage (I/O size and request/response frequency) ... are not considered.”

+ Even if the size of input and output is protected, the adversary will still learn the fact that the event occurred, is this considered a leakage under your threat model?

**Author response:** This kind of information leakage is not considered in our work**.** We do not defend against this because the adversary does not learn which type of the event occurs.

- How does Aurora hook on the events of output requests by an enclave (Section 3B)?

**Author response:** The output requests are forwarded by a kernel thread (ashmd) as described in Section 3D. This thread only handles exceptions (SMIs from an enclave to SMVisor, or IRQs from SMVisor to an enclave), and learns nothing about the message in plaintext. Denial of services attack such as drop or delay is not considered.

- The workflow in Section 3B can largely be moved to the figure caption. Try to keep the high-level details in the main text. In that text, specify that the driver is UEFI driver.

**Author response:** Good point! We move the workflow into the caption of Figure 2.

[HELP] This makes the caption too long to read !

- Section 3E, Aurora seems to be susceptible to time-of-check to time-of-use (TOCTOU) attack, if not, please clarify why.

**Author response:** When SMM is entered, the kernel is suspended. Aurora first detects if time attack has been launched, if not, it retrieves the time value and returns. TOCTOU happens when the timing of check and use can be separated. Aurora leverages SMM which guarantees that the operation of check and use are atomically performed. We complement this at the last paragraph in Section 4C now.

- Consider renaming disconnection to termination in Section 3F.

**Author response:** Done.

- Consider renaming Section 3G to 'Security Enhancement and Performance Optimization'.

**Author response:** Done.

- Is nonce used for encryption (Section 3G)?

**Author response:** Yes. Nonce is used for replay-proof; we add this to Section 3F: “We use a constant-time AES-128-GCM algorithm to defeat cache timing attacks. Nonces are used for replay proof.”

- For data obliviousness, if the blocks are of 4KB, they will have no random padding. All 4K size blocks which happen to have the same content will still leak. Please double check your scheme and ensure this attack is not possible.

**Author response:** Thank you for pointing out this subtle attack. We may force to split a 4KB message into at least 2 pieces and add padding to prevent possible attacks: “The maximum length of a message is 4080 Bytes, leaving at least 16 bytes for random padding.”

- Claims of Aurora solving phishing attacks in Section 4A are wrong. Please use a different example.

**Author response:** Thank you. We rephrase those sentences and focus on the keylogger attack now.

- The technique of using a pre-shared secret melody has a better entropy than the previous scheme, but the paper should clarify that it assumes that this pre-shared secret is not known by the adversary (Section 4B).

**Author response:** We clarify that “Since the melody is a pre-shared secret only known by SMVisor and the user, it is impossible for the adversary to learn and mimic this process.”

- How does aurora detect that the hardware clocks are being altered in a consistent way (Section 4C)?

**Author response:** In Section 4C, we use an SGX counter thread to provide a baseline value for hardware clocks. Because its counter value cannot be modified as it is inside SGX, when hardware clocks are altered consistently, the enclave thread can infer an attack due to inconsistency between the counter value and hardware clock value.

- In Section 4D, Can you list which all other storage systems Aurora can be applied to?

**Author response:** We reclaimed in Section 4D, “Note that we believe that Aurora’s approach is also suitable to support for other storage systems such as SATA HDD, NVMe SSD, etc.”

- Consider combining Section 5B and 5C

**Author response:** Good idea. Done.

- Limitations for trusted display are understandable, but can you check how Fidelius achieves/ side-steps this?

**Author response:** Fedelius introduces a Raspberry Pi and a full stack of system software, which bypasses the host stack, for trusted input and display. We discussed about this in related work, i.e., section 8B.

- Add one more column to show the percentage slowdown in Table 6

**Author response:** Sure. Done.

- Do not make any claims about how Aurora has side-channel protection in certain cases (Section 6B) if it is explicitly out of the scope of the paper.

**Author response:** Yes, you're right. We have removed those claims in Section 6B.

- Consider adding a comparison table with checks and crossed to clarify design goals and choices with respect to related work such as ROTE, can be added to the evaluation.

**Author response:** Cool, that is indeed a related work. We have added it into the Table V.

- Clarify how the storage deception attack detects denial-of-service if the adversary drops requests, if it is not possible, please scope it out.

**Author response:** We do not consider the denial of service attack to Aurora’s framework. As long as the framework works correctly (the kernel thread can forward SMI and SMI can not be blocked), denial of service to the storage service (drop requests) can be detected. We clarify our assumptions in the Threat Model and Storage Deception Attacks (Section 6B).

- Clarify the reasons why Aurora anticipates overhead ratio to decrease in WAN environment (Section 7B), or remove the claim.

**Author response:** We reclaimed in Section 7B, “We anticipate the overhead ratio to decrease in a WAN environment, because the delay of a geographically distributed public network should be larger than our LAN setting.”

- Clarify which types of SQL queries were executed (e.g., select, insert, delete), how many of them failed due to configuration challenges (Section 7C).

**Author response:** We re-clarify them in Section 7C.

- Last para in Section 8A is not readable due to awkward English, please consider cleaning it up.

**Author response:** We have rewritten Section 8A by removing fancy terms and details, and rephrasing in a simple way.

- Minor language and presentation issues

+ move table 3, 4, 5 to page 8 for ease of reading. better to position the tables where they are first referred to in the text.

**Author response:** Thanks for your kindly reminder. We have positioned these tables to appear before the referred text.