# CCC SIG Proposal: Attestation

Version 1.1
11 March 2021
Associated Technical Charter: Draft (Ready for Submission)

## Summary

The purpose of this document is to propose a special interest group for the Confidential Computing Consortium that would focus on *interoperability* between different types of Confidential environments, as well as between Confidential and non-Confidential environments, focused on Authentication and Authorization flows underpinned by Attestation.

## Current Landscape

There are currently several confidential computing frameworks available to customers: Intel SGX SDK for Linux and Windows, Google Asylo, Open Enclave SDK, Fortanix EDP, Enarx, Google Cloud Confidential Computing, Azure Confidential Computing and so on. One aspect of all of these frameworks is relying on attestation quotes/evidence from the underlying Trusted Execution Environments. Attestation is used for cryptographic validation of both the workload's and TEE's provenance and features, allowing confidential applications to "bubble up" proof of their protection properties to customers/auditors. In addition, multiple frameworks (e.g., Open Enclave SDK, Enarx, Google Asylo) and projects presented in the Confidential Computing Consortium (CCC) offer a way to authenticate and authorize applications running within confidential computing environments (including mutual authN/authZ based on attestation). The same capabilities are used to exchange secrets and to establish secure connections based on the result of the Attestation processes that are subject to a policy that customers and infrastructure owners can apply.

## Need for Common Standards

Attestation features found in current offerings in the Confidential Computing space meaningfully improve security of confidential applications by offering computing parties a way to verify specific properties of each other's workloads, lessening the need for blind trust that must be placed on the other party. However, these solutions do not yet offer a standard way for customers to declare, configure and authenticate attestation claims of a peer workload that might be running in a peer TEE (e.g., Intel SGX, AMD SEV, ARM TrustZone, etc. or other

protected environment), they may offer partial or limited framework-independent *and* platform-independent ways to accomplish Attestation interoperability goals. Additionally, they do not have built-in support for web services and application frameworks popular among the user community.

Therefore, it is of significant interest to the Confidential Computing Consortium to explore how harmonisation and de-fragmentation can be achieved (i.e., by producing interoperable implementations based on real-world use cases).

# SIG Attestation Topics

1. Standards-based interoperable formats for cross-platform representation of confidential computing claims, including platform-independent or application-level information and platform-dependent or TEE-specific information from multiple TEE providers. This includes reviewing and providing feedback to SDOs on their existing draft specifications to be compatible with the CCC requirements.
2. Requirements for the TEE-independent method that would allow applications to present confidentiality claims (including none for non-confidential applications), to other computing parties and to verify them, utilizing industry standard formats like X.509, COSE, transport protocols like TLS, EAP, message encodings, and industry standard tools like OpenSSL and MbedTLS.
3. Discussions and recommendations for the actual claims to include into attestation, including claims about TEE's operating context, chain of trust, etc, and ways to link different types of claims for the same platform (referred by the RATS architecture as Endorsements/Reference Values).
4. A list of any unsolved problems that require further research and/or possible collaborations with academic institutions.

# SIG Attestation Deliverables

1. Design specification for items (1) and (2) from the list in the Topic section, with a reference implementation of data structure and validation methods.
2. Recommendations for binding of the platform/firmware/boot layers of attestation to the 'runtime' identity of a Confidential Computing workload to represent "continuous confidentiality" notion. For example, deciding whether or not proof of possession protocols (e.g., RFC 8747) will satisfy CCC use cases or scenarios.
3. Recommendations for abstracting the problem of TEE verification to be compatible with multiple TEE vendors, with open-sourced proof of concept (POC) implementations.

4. Libraries and “bindings” for popular programming languages and web frameworks that would allow users to offer easy to use and *declaratively* configure ways to carry out Attestation processes in applications using their confidential computing frameworks without requiring users to write complex code.
5. Attestation service requirements, including policy support, for Cloud Providers as cloud specific services offering “for-hire” attestation services for first-party and third-party workloads.
6. Organize periodic virtual or physical (Open Source Summit, CCC meetings, IETF hackathons, etc.) interoperability testing events or hackathons where different POC implementations can be used and usability and feature improvements can be proposed.

# Success Criteria

1. A successful POC would effectively reuse existing mechanisms (protocols, tools, data formats) as much as possible. If there are gaps found in existing standards, the SIG would focus on interoperability of POC implementations and refer proposed changes to SDOs.
2. A successful proposal would describe a set of requirements and provide POC implementations demonstrating interoperability between as many CCC projects as practical. For example, two confidential applications using different frameworks/SDKs and different underlying TEEs might need the ability to mutually authenticate and attest each other either (including, when needed, their environments) directly or via Proxy services, and then exchange sensitive data.
3. A successful proposal would ideally be compatible with or incorporate/”wrap” relevant standards such as (IETF RATS Working Group Remote Attestation Specifications and IETF TEEP Architecture). A successful proposal would be complemented with a technical design and a recommendation for how it can be implemented in popular SDKs and web frameworks in order to minimize a need for custom code. A POC implementation should include a sufficient number of code samples covering popular languages and web frameworks or encapsulated as a part of an OSS project.
4. A successful proposal would include a way for applications to *declaratively configure* the confidential computing state (or *negotiable* range of expectations) they expect their computing partners to satisfy, to be enforced and verified either by the underlying frameworks or by dedicated services.

# Minimum Viable Governance

1. 5 chairs, nominated to represent different companies that are committed to this work:
    - Arm: Thomas Fossati - thomas-fossati (github)
    - Google: Keith Moyer - KeithMoyer (github)
    - Microsoft: Aeva Black - @AevaOnline (github)
    - Intel: Shanwei Cen - @shnwc (github)
    - Red Hat: Mike Bursell - MikeCamel (github)

2. Slack Channel for ad-hoc communication and announcements: confidentialc-0kw1467.slack.com.
3. Meetings
    - Meeting Space: Zoom.
    - Meeting Minutes: Google Doc
    - Meeting Cadence: Every 2 weeks, as long as there is a proposed agenda.
    - Agenda items will be gathered in the Google Doc and can be proposed by anyone in CCC. The agenda and minutes link will be pinned in the same slack channel.
4. “Bias towards action”: any attendee in the SIG (not merely chairs) can put topics on the agenda, bring discussions forward, weigh in with opinions, and advance code (ranging in from POC to production-ready libraries and components) .
5. Status Updates: Monthly to start with. Chairs have a rotating responsibility to publish status update documents in the Github repo below.
6. Github Organization ***CCC-Attestation***RATS for documents, proposals and prototype code (in separate repos), as well as work item tracking (via Github Issues and Github project boards).
    - Agreement/TAC approval for recommended license: Apache 2.0.
    - Each POC can bring their own different license if necessary, subject to SIG chairs approval by a simple majority vote.
7. Mechanisms for adding deliverables:
    - An open Issue in Github Repo describing a problem and a proposed solution sketch.
    - “Chairs ratify”: voting by chairs is used to ratify a decision after a loose consensus is reached within the overall group.
    - First proposed deliverable that will be subject to this mechanism: Design specification for items (1) and (2) from the list in the Topic section, with a reference implementation of data structure and validation methods.
8. SIG chair responsibilities are primarily administrative, lightweight, and may rotate between chairs by election or delegation. For example:
    - Providing periodic reports to the TAC
    - Keeping public minutes and maintaining the agenda
    - Ensuring any work items are appropriately tracked in some agreed-upon fashion
9. After 2 years, plan to hold chair elections as follows:
    - TAC members nominate (or re-nominate) SIG chair candidates.
    - Any time a SIG chair chooses to step down, there will be a nomination period by the TAC to fill the vacancy. Nominations will be open to any active member of the SIG. This will not be a public vote.
    - A SIG chair may be removed by the TAC if inactive for more than three months. A SIG chair may delegate responsibilities to another to temporarily stand in (e.g., during a period of leave) without releasing their seat.
    - At the end of two years, two of the seats will come up for the general election. Any member of the SIG who has been active consistently over the prior 12 months may run for a seat. The term length for these seats will be two years. The

electorate will be made up of all contributing members during the prior 12 months.
    - One year later, the alternate three seats will come up for the general election, under the same terms, membership body, and electorate.
10. The governance structure may be amended at any time by simple majority (3/5) vote of the SIG chairs and with approval from the TAC.

# Future Roadmap

Agreement of base interoperability requirements and implementations will allow the SIG to extend guidance to Confidential Computing practitioners and offer further motivation focusing on these scenarios:

- TEE Vendor and Cloud Provider "Joint Attestation" Standard that extends attestation claims to cover ownership and location of a compute resource that is hosting a TEE environment.
- Secret Sealing and Key Escrow Standard focusing on hybrid cloud, on-premise and multi-cloud operations.
- Multi-Party Compute Standard focusing on ability for mutually distrustful computing parties to build applications that work on common/joint sets of data.
- Verification tools, such as Testing Suite, usable for both CCC and for the third-party auditors, including regulatory compliance.