Remote ATtestation ProcedureS
Internet-Draft
Intended status: Standards Track
Expires: 6 January 2026

D. Wang
F. Liu
Huawei

5 July 2025

Distributed Remote Attestation
draft-wang-rats-distributed-remote-attestation-00

Abstract

In the RATS architecture, remote attestation typically involves multiple roles: verifier, attester, endorser, reference value provider, and relying party. However, in complex networks, the large number of entities in each role can significantly increase the cost of communication and computational resources. This document proposes a simplified remote attestation method based on distributed ledger(DL) technology, which aims to reduce the overhead for participants in multi-party networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 6 January 2026.

Copyright Notice

Table of Contents

1. Introduction

In the current remote attestation procedures, the attester needs to demonstrate the freshness of the evidence, which is achieved by including some fresh data within the evidence. For instance, RFC 9334 [RFC9334] proposes may use a random number provided by the verifier/relying party , or an epoch ID sended by a distributor as the freshness proof. Since the freshness proofs are generated by a specific verifier or distributed for a specific verifier, the AE generated based on these freshness proofs cannot be verified by other verifier. This means that the computational results cannot be reused. For instance, when the attester needs to establish communication with multiple verifiers, even if the AE is highly similar, the attester still has to compute multiple AEs based on each verifier's challenge which cost multiple computational resources.

To address this issue, it is reasonalble to introduce a trusted institution that can periodically provide unpredictable public random to the attester for generating a pulic AE that could be verified by multiple verifiers. This document proposes a method that uses the hash of the latest block in DL as a public challenge(PC). Since the DL is maintained by multiple participants in the network based on a consensus mechanism, it is trustworthy, and the hash of the block is fresh because it has a timestamp that proves its generation time. Thus, this hash can serve as a PC for the attester, eliminating the need for the attester to repeatedly compute based on each verifier's challenge thereby simplifying the computational load of the attester and the communication between the verifiers and the attester.

In addition to serving as a PC provider, DL can offer more services for RAT. For example, it can act as a secure and trustworthy platform for information sharing. Reference value providers and secure hardware platform manufacturers (endorsers) can publish their reference values (RV) or public keys (PK) to reduce the workload for verifiers to obtain them one by one. Furthermore, based on the DL, verifiers can publish their attestation results (AR), allowing other verifiers or relying parties to reuse these ARs.

Furthermore, smart contracts can be deployed on DL to execute AE verification functions, which is more trustworthy compared to single verifier verification. When a smart contract is executed, the execution result must be agreed by the majority of nodes before it is confirmed. Even if some nodes are compromised/attacked, other nodes can still ensure continuous and accurate verification, avoiding the impact of single-point failures. Finally, the smart contract publishes the verified results on the DL for use by the relying parties.

Based on the above research, this document describes various combination patterns of the above three processes and existing processes, there are three modes: on-chain-based publication/ attestation and off-chain-based verification, on-chain-based publication and off-chain-based attestation/verification, and on-chain-based publication/attestation/verification.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Conventions and Definitions

On-chain-based Publication: In DRA, this refers to the process for publishing key data (e.g., PK and RV) used for verification or the process for publishing verification results (AR).

On-chain-based Attestation: In DRA, this refers to the process in which an attester generates AE using the DL hash as the challenge.

Off-chain-based Attestation: In DRA, this refers to the process in which an attester generates AE via traditional methods, such as responding to a challenge sended by the verifier or relying party.

On-chain-based Verification: In DRA, this refers to the process in which a smart contract deployed on the DL verifies AE.

Off-chain-based Verification: In DRA, this refers to the process in which a verifier verifies AE using conventional methods.

4. DRA patterns

4.1. on-chain-based publication/attestation and off-chain-based verification

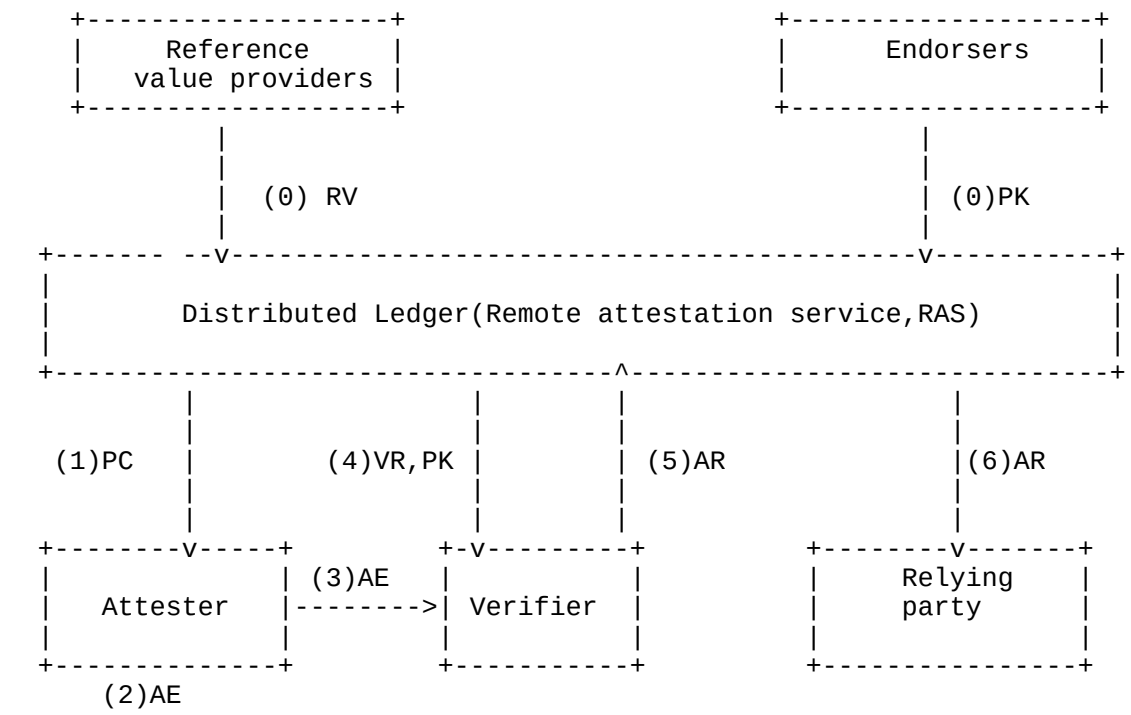

Figure 1 on-chain-based publication/attestation and off-chain-based verification

As shown in Figure 1, one mode of DRA is on-chain-based publication/ attestation and off-chain-based verification.

0) Registration: Reference value providers and endorsers register RVs and PKs with the DL.

1-2) AE Generation: The attester retrieves the latest block hash from DL as a challenge to generate the AE. By leveraging the immutable timestamp recorded in the block header of the DL, the generated AE is inherently bound to the exact time of its creation, ensuring verifiable freshness. The attester can initiate this process either in response to a request from the verifier/relying Party or proactively through an autonomous trigger mechanism.

3) AE Transmission: The attester sends the AE to the verifier. The attester can reuse the same AE with multiple verifiers if need.

4) AE Verification: The verifier queries the DL to retrieve the RV, PK and check the freshness of the challenge and verify the AE.

5) AR Generation: The verifier generates an attestation result after a successful verification, then publishes it on the DL.

6) AR Acquisition: The Relying Party can directly query and obtain the AR from the DL.

4.2. on-chain-based publication/attestation/verification

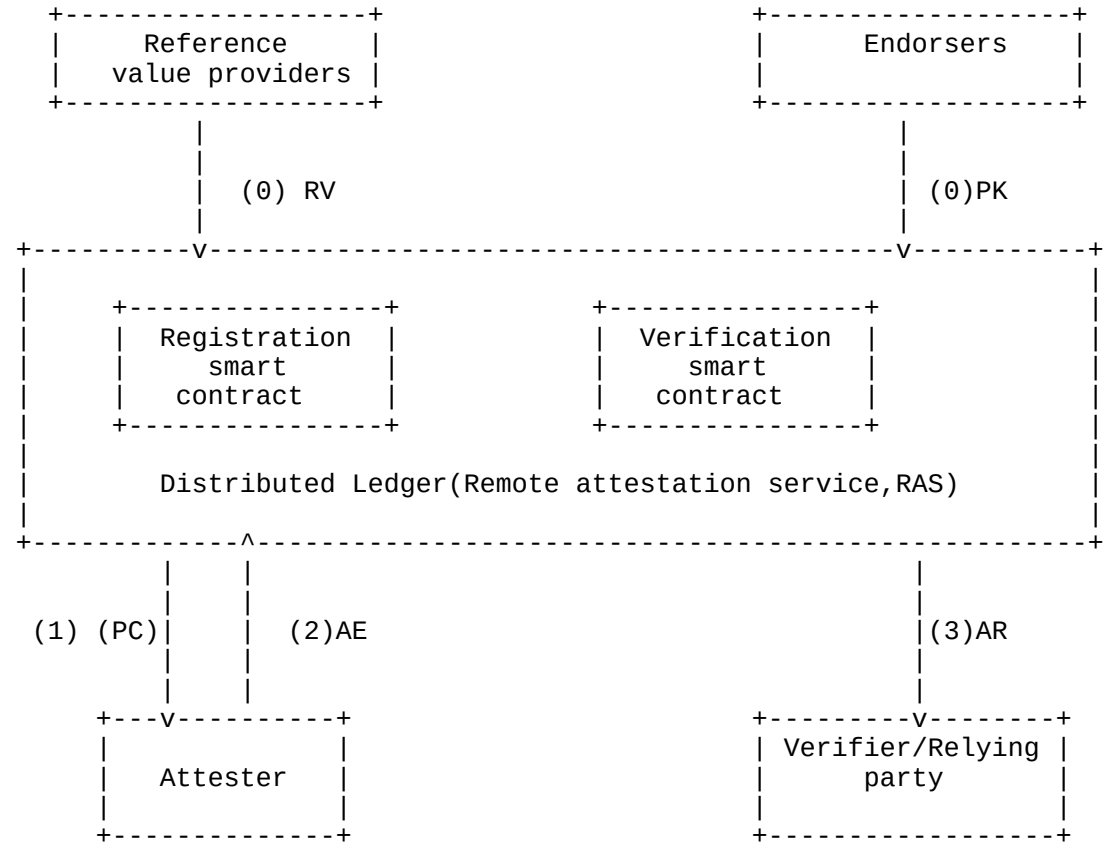

Figure 2 on-chain-based publication/attestation/verification

As shown in Figure 2, the second mode of DRA is n-chain-based publication/attestation/verification.

0) Registration: Reference value providers and endorsers register RVs and PKs with the DL.

1-2) AE Generation and Verification: The attester retrieves the latest block hash from the DL as a challenge, generates the AE, and publishes it to the DL by invoking the registration smart contract. Subsequently, the verification smart contract——triggered either by the Verifier or the registration smart contract——executes the AE verification process.

3) AR Acquisition: The relying party can directly query and obtain the attestation result from the DL.

4.3. on-chain-based publication and off-chain-based attestation/ verification

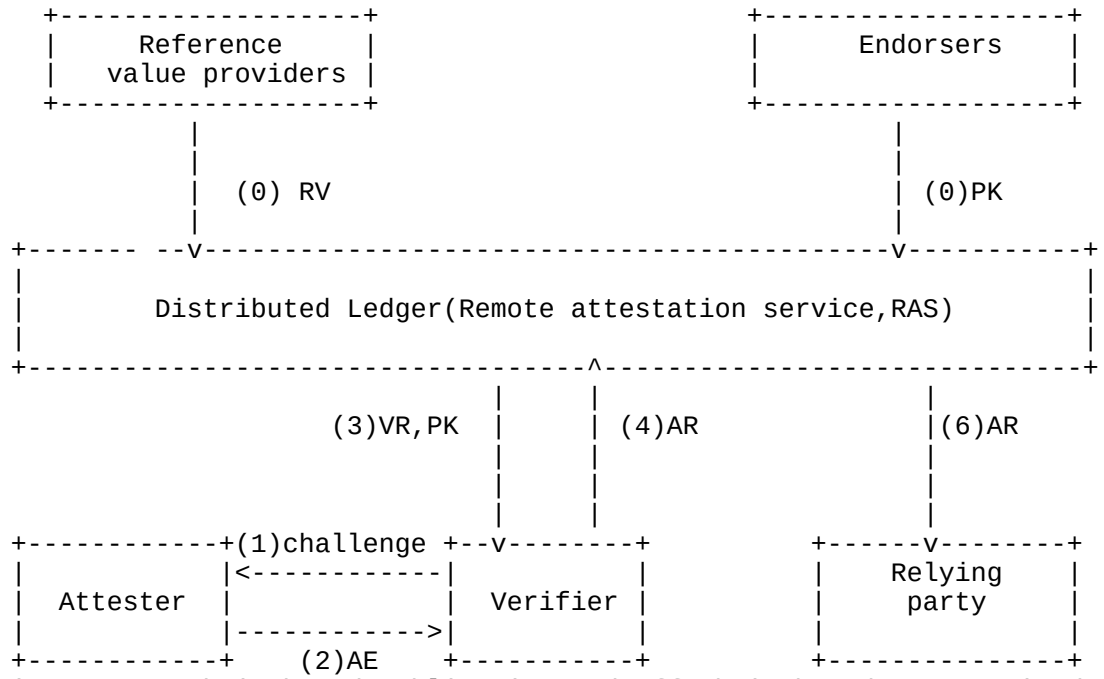

Figure 3 on-chain-based publication and off-chain-based attestation/verification

As shown in Figure 3, the third mode of DRA is on-chain-based publication/attestation and off-chain-based verification.

0) Registration: Reference value providers and endorsers register RVs and PKs with the DL.

1-2) AE Verification: Same as the existing verification process, the verifier sends a challenge to the attester. In response, the attester generates the AE and submits it directly to the verifier.

3) AR Generation: After an successful verification, the verifier generates the AR.

4) AR publication: Verifier publishes AR to the DL.

5) AR AcquisitionThe relying party can directly obtain the result from the DL.

5. Discussion

5.1. Unpredictability of PC

The block generation mechanism of DL is driven by a consensus algorithm among nodes. Nodes package transaction data based on established algorithmic logic, forming a block that includes the hash of the previous block, a timestamp, and the current set of transactions. After being verified by the majority of nodes in the DL network, the block is added to the DL. Due to the uncertainty of timestamps and transaction sets, DL nodes cannot predict the hash of the next block in advance. Moreover, because of the one-way feature of hash functions, attackers cannot predefine a hash and reverse-engineer or construct the original data to generate the desired hash. These characteristics collectively determine the uniqueness and unpredictability of block hashes, thereby ensuring the security of PC.

5.2. Freshness Analysis

For the DRA service, the blocks of the distributed ledger (DL) need to be generated at appropriate time intervals, such as every few seconds. The consensus rules trigger the new block generation process periodically through preset time parameters. Even if there is no transaction data within a specific period, nodes will still generate an empty block containing basic information like the hash of the previous block and timestamps according to established algorithms. This approach aims to maintain the continuity of the DL chain structure and the orderliness of timestamps, thereby ensuring the freshness and validity of PC.

6. IANA Considerations

This document has no IANA considerations.

7. Security Consideration

TODO

8. References

8.1. Normative Reference

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

8.2. Informative References

[RFC9334] Thaler, D., Richardson, M., Smith, M., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", January 2023, <https://datatracker.ietf.org/doc/rfc9334/>.

Acknowledgments

TODO

Authors' Addresses

Donghui Wang
Huawei
Email: wangdonghui124@huawei.com

Faye Liu
Huawei
Email: liufei19@huawei.com