GroupTEE: Efficient Group-Oriented Secure Channel Establishment for TEEs

Jinghui Zhang[1,2,3], Hai Liu[4], Wenlong Kou[1,2,3,✉], Fenghua Li[1,2,3], and Yunchuan Guo[1,2,3]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] State Key Laboratory of Cyberspace Security Defense, Beijing, China
[4] College of Computer Science and Technology, Guizhou University, Guiyang , China
kouwenlong@iie.ac.cn

Abstract. Trusted Execution Environments (TEEs) enable secure cross-domain collaborative computation. However, establishing group-oriented secure channels among mutually distrustful enclaves remains challenging, as it must provide mutual authentication while preserving group-level security semantics. To address this, we present GroupTEE, a group-oriented secure communication framework for TEEs. GroupTEE unifies CA-certified identities and remote attestation into a trust-binding mechanism that cryptographically binds group membership and enclave integrity to the derived group session key within a single protocol execution. We prove in the ROM that GroupTEE achieves EUF-CMA authentication, session-key indistinguishability, and key consistency. Experiments show that GroupTEE reduces per-participant attestation generation operations from $O(n)$ in pairwise baselines to $O(1)$ per session.

Keywords: TEEs · Remote Attestation · Secure Channel · Confidential Computing · Authenticated Group Key Agreement.

1 Introduction

With the growing adoption of confidential computing, TEEs have become a practical foundation for secure cross-domain collaborative computation [14]. TEEs provide hardware-enforced isolation and remote attestation, enabling privacy-sensitive collaboration among mutually distrustful parties while protecting data in transit and confining execution within enclaves. A representative example arises in collaborative automotive manufacturing, where an original equipment manufacturer distributes large-scale CAD blueprints to multiple suppliers, each authorized to access only a role-specific portion. As these artifacts are data-intensive and frequently updated, traditional pairwise secure channels incur substantial communication overhead. A more scalable approach is to encrypt the blueprint once inside the sender-side TEE using a group session key and disseminate it to all authorized parties, where each receiver decrypts within its local TEE and enforces identity-based fine-grained access control.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17

Such scenarios illustrate a broader requirement for group-level trust semantics, in which enclaves collectively establish a cryptographically bound trust relationship under a unified attestation framework. However, because enclaves may run in mutually distrustful administrative domains, an efficient group-oriented secure channel establishment framework is essential to achieve resilience against impersonation, replay, and man-in-the-middle attacks.

Motivation. Considerable efforts have been devoted to establishing secure channels between mutually distrustful enclaves. Existing TEE-based solutions predominantly rely on pairwise constructions, extending to multi-party settings through repeated remote attestation and independent key-agreement executions. Although such pairwise compositions achieve mutual authentication, they incur redundant attestation operations and fail to provide a native abstraction for group-oriented secure channel establishment.

To address multi-party collaboration more broadly, Authenticated Group Key Agreement (AGKA) protocols have been extensively studied, enabling multiple participants to establish a common group session key while ensuring mutual authentication. Unlike TEE-specific solutions, AGKA protocols provide a hardware-agnostic foundation, offering valuable insights for designing efficient and secure multi-party schemes in untrusted environments. However, directly applying existing AGKA schemes to multi-party TEE collaborations is insufficient due to the following challenges.

(1) *Lack of Identity–Attestation Binding.* Conventional AGKA protocols do not integrate hardware-backed remote attestation into the authentication process. Consequently, they cannot ensure that key derivation and message processing occur within verified TEE instances. In TEE-based settings, where security depends on both cryptographic identity and enclave integrity, the lack of identity–attestation binding leads to incomplete trust establishment.

(2) *Insufficient efficiency in resource-constrained TEEs.* Resource-constrained TEEs impose strict limits on memory and computational capacity within secure enclaves. At the same time, security guarantees must remain robust against potentially malicious receivers. Achieving an effective balance between efficiency and security in multi-party TEE settings thus presents a significant challenge.

Contribution. To address these challenges, we propose GroupTEE, a group-oriented secure communication framework for TEEs. GroupTEE integrates identity authentication and hardware-backed attestation into a unified group-level trust establishment mechanism, while reducing the overhead inherent in pairwise designs. Our main contributions are summarized as follows.

(1) To our knowledge, we are the first to formally define the notion of group-oriented secure channel establishment for TEEs and to present GroupTEE, a protocol that enables collective trust establishment among multiple enclaves within a single protocol execution. In contrast to conventional pairwise compositions, GroupTEE models the group as a unified security entity and natively enforces group-level trust semantics. Furthermore, it integrates identity authentication

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17

and remote attestation into a unified cryptographic trust-binding mechanism, ensuring that the set of authenticated enclaves, their attested integrity, and the derived group session key are securely bound to the same authenticated session context.

(2) GroupTEE eliminates redundant pairwise attestation procedures by allowing each participant to generate attestation evidence only once per session. It avoids repeated channel establishment and reduces authentication complexity compared with traditional pairwise approaches. As a result, GroupTEE significantly lowers computational and communication overhead in group-oriented deployments.

(3) We provide a formal security analysis in the Random Oracle Model (ROM), proving that GroupTEE achieves Existential Unforgeability under Chosen Message Attack (EUF-CMA) security for authentication, session-key indistinguishability and key consistency. We further validate GroupTEE correctness using BAN logic and symbolic verification in Scyther under the Dolev–Yao adversary model. In addition, we implement GroupTEE and evaluate its computational and communication overhead on representative TEE platforms. The results demonstrate that it is efficient and practical for multi-party deployments.

2 Related Work

2.1 Authenticated Group Key Agreement

Authenticated group key agreement protocols constitute a fundamental cryptographic framework for secure group communication. Existing research focuses on improving efficiency and scalability. Constant-round constructions [18, 20] minimize communication latency by bounding the number of protocol rounds independently of group size, making them particularly suitable for large but relatively static groups. Tree-based dynamic protocols, such as TGDH [8], structure participants into logical key trees, reducing the computational and communication overhead of join and leave operations from $O(n)$ to $O(logn)$. More recently, continuous group key agreement protocols, exemplified by TreeKEM [1, 16] , emphasize forward secrecy and post-compromise security through continuous key evolution and have been widely adopted in modern secure group messaging systems.

These protocols natively support group semantics by enabling collective establishment of a shared group session key. However, their security guarantees are defined at the cryptographic abstraction layer and typically assume pre-established authentication, without binding protocol execution to hardware-rooted trust. Consequently, in the absence of remote attestation, they provide no cryptographic assurance that key derivation and message processing occur within attested enclave environments when deployed in TEE-based systems.

2.2 Remote Attestation

Recent research has explored the integration of remote attestation with established secure communication protocols to provide assurance that encrypted

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17

channels terminate within attested TEEs. Approaches such as TLS+RA [19], RATLS [17], and HTTPA [9, 10] embed attestation evidence into the Transport Layer Security (TLS) [15] handshakes, preserving two-party channel security while binding endpoints to an attested enclave. Similarly, Majumder *et al.* [11] combine Diffie–Hellman key exchange with SGX remote attestation for attestation-based endpoint authentication. Beyond client–server models, distributed TEE systems such as NTEE [3] and MultiTEE [13] enable attested secure channels between TEEs, typically established through mutual attestation and Diffie–Hellman based authenticated key agreement.

However, these approaches rely on pairwise trust establishment as the primary security primitive. In fully connected group settings, each participant must establish $n-1$ independent secure channels, often incurring redundant attestation generation and verification overhead. Consequently, while these schemes enhance endpoint authenticity, they do not offer a native group-level trust abstraction or direct group key establishment.

3 System and Security Model

System Model. The system model is illustrated in Fig. 1. It consists of a trusted third-party Certificate Authority (CA) and a set of participants $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$. The system targets cross-domain data sharing scenarios in which mutually distrustful entities from different administrative domains establish mutually authenticated secure channels between their TEEs prior to exchanging sensitive data. The roles of each entity are described below.

(1)*CA.* The CA is responsible for system initialization and participant registration. It verifies remote attestation evidence of participants' TEEs and issues certificates that bind identities, long-term public keys and platform measurements.

(2)*Participants.* Each participant P_i is equipped with a TEE that provides hardware-enforced isolation and remote attestation capabilities. All sensitive operations and cryptographic materials, including long-term secrets and group session key, are generated and protected within the TEE.

Communication among participants takes place over an untrusted network. For protocol execution, participants are logically arranged in a cyclic order to facilitate mutual TEE authentication and secure channel establishment, while the underlying physical network topology remains unrestricted.

Threat Model. We consider an active adversary operating in an untrusted network and system environment. The adversary may fully control the communication network and the untrusted host software, including the operating system and application layer. It can launch active attacks including message forgery, identity impersonation, replay, and man-in-the-middle attacks, and may attempt to extract secret information via host-level compromise. However, the adversary cannot directly access or extract secrets protected within correctly functioning TEEs.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17

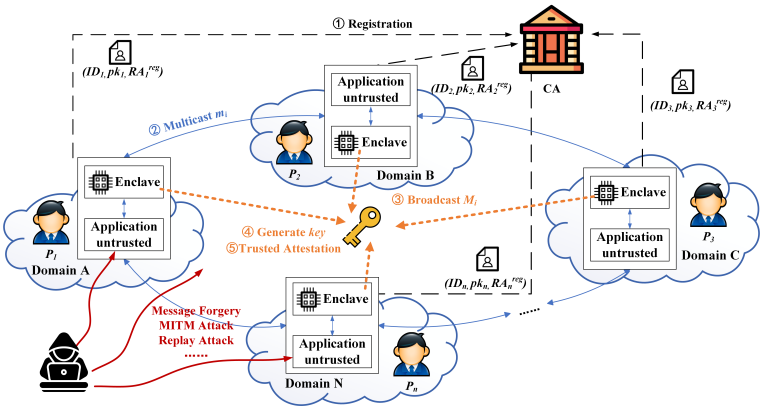


Fig. 1: System architecture and threat model for GroupTEE

Security Assumptions. The security of GroupTEE relies on the following assumptions.

(1) *Trusted CA.* The CA is trusted to correctly verify remote attestation evidence and to issue certificates binding participant identities, long-term public keys and platform measurements. The CA does not collude with adversaries.

(2) *Trusted TEE Execution.* Participants may belong to mutually distrustful administrative domains . However, each participant P_i is assumed to be honest and non-colluding, executing GroupTEE faithfully inside its TEE. The TEE hardware is trusted and uncompromised, ensuring isolation, secure storage, and remote attestation. Cryptographic primitives are assumed secure, and adversaries may compromise the OS or application layer but cannot break TEE-enforced guarantees.

Secure Goals. Under the above model and assumptions, GroupTEE is designed to achieve the following security objectives.

G1: Channel Endpoint Correctness. Secure channels are cryptographically bound to verified enclaves, preventing an adversary from offloading channel termination to an unauthorized environment.

G2: TEE Mutual Authentication. GroupTEE supports mutual authentication by allowing participants to verify both the identities and execution environments of their counterparties.

G3: Session Key Confidentiality and Consistency. The group session key must remain confidential against any adversary not compromising the TEE hardware. All honest participants in the same session must derive an identical key.

G4: Session Independence and Freshness. Sessions are isolated to ensure replay resistance and session independence. Compromise of a single traffic key does not enable derivation of traffic keys from other sessions.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17

4 Proposed Scheme

The GroupTEE protocol proceeds in six phases: (i) system initialization, (ii) registration, (iii) group context setup, (iv) authenticated group key agreement (AGKA), (v) attestation-based trust binding and group key confirmation, and (vi) traffic-key derivation and channel activation. The notations used throughout the protocol are summarized in Table 1.

4.1 System Initialization

Before protocol execution, the CA initializes the system public parameters, which are fixed for all participants and sessions. Specifically, the CA selects an elliptic-curve group $\mathbb{G}$ of prime order q with generator G, a hash function $\mathsf{H} : \{0,1\}^* \to \{0,1\}^\lambda$, and a key-derivation function $\mathsf{KDF} : \{0,1\}^* \to \{0,1\}^\kappa$. We use $\mathsf{Encode}(\cdot)$ to denote a canonical encoding of ID_{list}, and $\mathsf{EncodePoint}(\cdot)$ to encode group elements in $\mathbb{G}$ as bit strings.

Table 1: Notations Used in GroupTEE.

Notation	Definition
P_i, ID_i	Participant and long-term identity of P_i
(pk_i, sk_i)	Long-term public/secret key pair of P_i (generated in TEE)
$Cert_i$	CA-issued certificate binding (ID_i, pk_i, SM_i)
$\mathbb{G}, q, G$	Elliptic-curve group, its order, and generator
ID_{list}	Intended participant set for a session
sid	Fresh session identifier
$Token$	Session context token $\langle ID_{list}, sid, \sigma_{CA}\rangle$
σ_{CA}	CA signature on $(\mathsf{Encode}(ID_{list}) \parallel sid)$
$r_i, R_i = (R_{ix}, R_{iy})$	Fresh random scalar and ephemeral public point $R_i = r_i \cdot G$
s_i, t_i	Schnorr-style signature scalar of P_i for the first and second round
m_i, M_i	First-round message, second-round message
$X_i = (X_{ix}, X_{iy})$	Second-round point $X_i = r_i \cdot (R_{i+1} - R_{i-1})$
T_{-i}, R_{-i}, W_{-i}	Aggregates used to verify all M_j for $j \neq i$
K_i	Locally computed shared group element
GK	Group session key, defined as $\mathsf{KDF}(\mathsf{EncodePoint}(K_i))$
TK	Traffic key, defined as $\mathsf{KDF}(GK \parallel sid)$
$RA_i^{\text{reg}}, Msg_i^{\text{reg}}$	Attestation evidence for registration and message $\{ID_i, pk_i, RA_i^{\text{reg}}\}$
SM_i	Platform measurement asserted by remote attestation
$SessionHash_i$	Binding hash embedded in session-binding attestation
$RA_i^{\text{bind}}, Msg_i^{\text{bind}}$	Session-binding attestation evidence and message $\{ID_i, RA_i^{\text{bind}}\}$

4.2 Registration

Each participant P_i registers with the CA to bind its identity, long-term public key, and attested platform measurement, thereby obtaining a CA-issued certificate for subsequent sessions.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17

(1)*Registration Request.* Each participant P_i initiates registration by generating a long-term key pair (pk_i, sk_i) inside its TEE, where the secret key sk_i is sealed and non-exportable. The enclave computes a binding hash $BindHash_i = \mathsf{H}(ID_i \parallel pk_i)$ and embeds it into the `ReportData` field of an attestation report. Using the attestation interface, P_i produces the attestation evidence RA_i^{reg} over this report and submits $Msg_i^{\text{reg}} = \{ID_i, pk_i, RA_i^{\text{reg}}\}$ to the CA.

(2)*Certificate Issuance.* Upon receiving Msg_i^{reg}, the CA verifies RA_i^{reg} and extracts $BindHash_i$ from the `ReportData` field and checks consistency with the submitted ID_i and pk_i. If all checks succeed, the CA issues $Cert_i$ that cryptographically binds ID_i, pk_i and SM_i; otherwise, it rejects the request.

4.3 Group Context Setup

In this phase, the CA establishes a session-specific trust context for the target group by issuing a session context token, which is subsequently validated by all participants together with the corresponding certificates.

(1)*Session Context Token Issuance.* For a target group session, the CA determines the intended participant list ID_{list} and computes $\mathsf{Encode}(ID_{list})$. It then samples a fresh session identifier sid and signs $(\mathsf{Encode}(ID_{list}) \parallel sid)$. The session token is $Token = \langle ID_{list}, sid, \sigma_{CA} \rangle, \sigma_{CA} = \mathsf{Sign}_{sk_{CA}}\big(\mathsf{Encode}(ID_{list}) \parallel sid\big)$.

(2)*Certificate and Session Context Validation.* Each participant P_i obtains $Token$ and the certificate set $\{Cert_j\}_{ID_j \in ID_{list}}$. It verifies each $Cert_j$ and σ_{CA} under the CA public key and extracts the corresponding public keys pk_j. If any verification fails, P_i aborts. Otherwise, it accepts $(\mathsf{Encode}(ID_{list}), sid)$ as the authenticated session context.

4.4 Authenticated Group Key Agreement

Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ denote the set of session participants, indexed according to the canonical order of identities in ID_{list}. Participants are logically arranged in a cyclic order such that $P_0 = P_n$ and $P_{n+1} = P_1$. In this phase, the participants execute a two-round interactive protocol to establish a shared group session key GK, as illustrated in Fig. 2.

Neighbor Exchange of First-round Messages. Each participant P_i generates a first-round message m_i inside the enclave and sends it to its adjacent members.

Step 1. The enclave generates a random number $r_i \in \mathbb{Z}_q^*$ and computes the corresponding ephemeral public point $R_i = r_i \cdot G = (R_{ix}, R_{iy})$. If R_i equals the point at infinity, the enclave regenerates r_i.

Step 2. The enclave computes signature $s_i = sk_i h_i + r_i \pmod q$, where, $h_i = \mathsf{H}(ID_i \parallel R_{ix} \parallel R_{iy} \parallel \mathsf{Encode}(ID_{list}) \parallel sid)$.

Step 3. The enclave outputs $m_i = \{ID_i, R_{ix}, R_{iy}, s_i\}$, which the application sends to P_{i-1} and P_{i+1}.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17

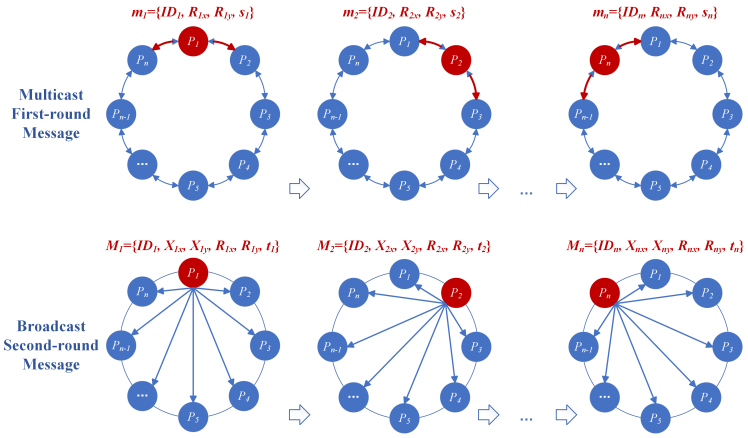


Fig. 2: Two-round communication process in AGKA.

Broadcast Second-round Message. P_i collects and verifies first-round messages m_{i-1} and m_{i+1} from neighbors, then generates second-round message M_i within the enclave, which is subsequently broadcast to all group members.

Step 4. The enclave verifies the received message m_{i-1} and m_{i+1} from the adjacent group members P_{i-1} and P_{i+1}, including the following procedures.

(1) Set $R'_{i-1} = (R_{(i-1)x}, R_{(i-1)y})$, $R'_{i+1} = (R_{(i+1)x}, R_{(i+1)y})$, and checks that both are valid curve points.

(2) Compute $h'_{i+1} = \mathsf{H}(ID_{i+1} \parallel R_{(i+1)x} \parallel R_{(i+1)y} \parallel \mathsf{Encode}(ID_{list}) \parallel sid)$ and $h'_{i-1} = \mathsf{H}(ID_{i-1} \parallel R_{(i-1)x} \parallel R_{(i-1)y} \parallel \mathsf{Encode}(ID_{list}) \parallel sid)$.

(3) Verify the following signature equations $s_{i+1} \cdot G \stackrel{?}{=} R'_{i+1} + h'_{i+1} \cdot pk_{i+1}$, $s_{i-1} \cdot G \stackrel{?}{=} R'_{i-1} + h'_{i-1} \cdot pk_{i-1}$. If any verification fails, P_i aborts.

(4) Compute $X_i = r_i \cdot (R'_{i+1} - R'_{i-1}) = (X_{ix}, X_{iy})$.

Step 5. Compute $t_i = \mathrm{sk}_i H_i + r_i \pmod q$ for the second-round message M_i, where, $H_i = \mathsf{H}(ID_i \parallel X_{ix} \parallel X_{iy} \parallel R_{ix} \parallel R_{iy} \parallel \mathsf{Encode}(ID_{list}) \parallel sid)$.

Step 6. The enclave outputs $M_i = \{ID_i, X_{ix}, X_{iy}, R_{ix}, R_{iy}, t_i\}$, which is broadcast by the application.

Derive Group Key. Each participant collects the second-round messages from others and verifies their authenticity. Upon successful verification, the group session key GK is derived within the enclave, as shown in Fig. 3.

Step 7. The application collects and stores second-round messages $M_{-i} = \{M_1, M_2, \ldots, M_{i-1}, M_{i+1}, \ldots, M_n\}$.

Step 8. Verification of the received messages proceeds as follows.

(1) The enclave samples a set of random weights $\{\alpha_j\}_{j \in \mathcal{P} \setminus \{i\}} \leftarrow \mathbb{Z}_q^*$ and outputs them to the application for aggregate computation.

(2) The enclave computes hashes $H'_{-i} = \{H'_1, H'_2, \ldots, H'_{i-1}, H'_{i+1}, \ldots, H'_n\}$, and, $H'_j = \mathsf{H}(ID_j \parallel X_{jx} \parallel X_{jy} \parallel R_{jx} \parallel R_{jy} \parallel \mathsf{Encode}(ID_{list}) \parallel sid)$.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17

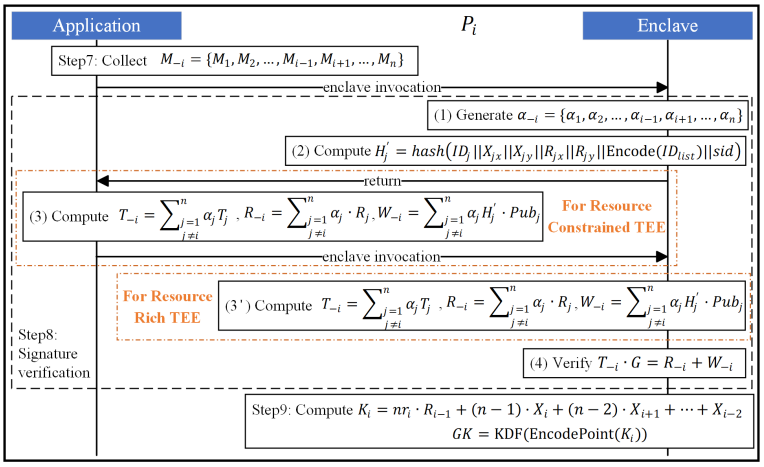


Fig. 3: Aggregate signature verification and group key derivation

(3) For resource-constrained TEEs, the application aggregates T_{-i}, R_{-i}, and W_{-i} and passes them to the enclave for verification. For resource-rich TEEs, the enclave can compute these aggregates internally. Specifically,

$$T_{-i} = \sum_{\substack{j=1 \\ j\neq i}}^{n} \alpha_j t_j, \quad R_{-i} = \sum_{\substack{j=1 \\ j\neq i}}^{n} \alpha_j R_j, \quad W_{-i} = \sum_{\substack{j=1 \\ j\neq i}}^{n} \alpha_j H'_j \cdot pk_j.$$

(4) The enclave verifies the equation $T_{-i} \cdot G \stackrel{?}{=} R_{-i} + W_{-i}$. If any verification fails, P_i aborts.

Step 9. Upon successful verification, the enclave computes the shared group element as $K_i = nr_i \cdot R_{i-1} + (n-1) \cdot X_i + (n-2) \cdot X_{i+1} + \cdots + X_{i-2}$. The group session key GK is then derived as $GK = \mathsf{KDF}(\mathsf{EncodePoint}(K_i))$.

4.5 Attestation-Based Trust Binding and Group Key Confirmation

Attestation-Based Trust Binding. After the AGKA phase, all participants have derived the same group session key GK inside their TEEs under a consistent session context (ID_{list}, sid). GroupTEE further performs a session-binding attestation phase to confirm that each participant executed the protocol inside an attested TEE for the current session.

(1)*Session-Binding Attestation Generation.* Inside its enclave, each participant P_i computes a fresh cryptographic session-binding hash $SessionHash_i = \mathsf{H}\big(ID_i \parallel \mathsf{Encode}(ID_{list}) \parallel sid \parallel GK\big)$, and embeds $SessionHash_i$ into the `ReportData` field of a local attestation report. The enclave then generates attestation evidence $RA_i^{\text{bind}} \leftarrow \text{CreateQuote}(\text{CreateReport}(SessionHash_i))$. Finally, P_i sends or broadcasts the trusted attestation message $Msg_i^{\text{bind}} = \{ID_i, RA_i^{\text{bind}}\}$ to the other verifiers.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17

(2)*Session-Binding Attestation Verification.* Upon receiving Msg_i^{bind}, the verifier first validates RA_i^{bind} using the attestation verification interface, including signature verification, certificate-chain validation, and checking that the reported measurement matches the expected reference value; otherwise, the protocol aborts. Upon successful verification, the verifier recomputes $SessionHash'_i = \mathsf{H}\big(ID_i \parallel \mathsf{Encode}(ID_{list}) \parallel sid \parallel GK\big)$ inside its enclave and checks whether it matches the value embedded in the `ReportData` of RA_i^{bind}. A match confirms that P_i derived GK inside an attested TEE for the current session.

Attestation Modes. GroupTEE supports different trusted attestation modes depending on the communication topology of the distributed application.

(1)*Broadcast Topology.* In the broadcast topology, where a distinguished sender or receiver disseminates information to multiple parties, GroupTEE supports sender-to-receiver attestation, receiver-to-sender attestation, and mutual attestation.

(2)*Group Communication Topology.* In the group communication topology, where all participants act as peers, GroupTEE adopts mutual attestation by default. Each participant P_i broadcasts its trusted attestation message Msg_i^{bind} and verifies the evidence RA_j^{bind} received from all other participants $P_j (j \neq i)$.

4.6 Traffic-key Derivation and Channel Activation

After successful trusted attestation verification, each participant derives a traffic key $TK = \mathsf{KDF}(GK \parallel sid)$ and activates the secure channel using an authenticated encryption with associated data (AEAD) scheme [12]. As shown in Fig. 4, in the broadcast topology, one domain communicates with multiple recipients; in the group communication topology, all domains form a fully connected communication graph. Data are protected as $(C, tag) = \text{AEAD}_{TK}(nonce, plaintext, AAD)$, where $AAD = (sid, ID_i)$. All cryptographic operations are performed inside the enclave.

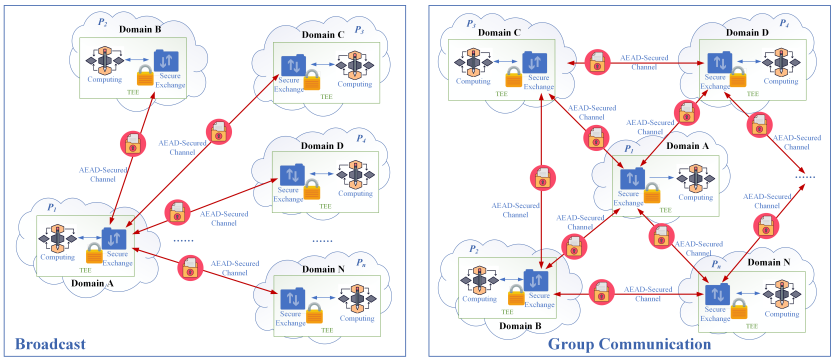


Fig. 4: Secure channel establishment in different topologies

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17

5 Security Analysis

We analyze the security of GroupTEE under the system and adversary model defined above and show that GroupTEE achieves the stated security goals.

G1: Channel Endpoint Correctness. GroupTEE guarantees channel endpoint correctness by cryptographically binding the derived group session key GK and session context (ID_{list}, sid) to each participant's attestation evidence. As GK is derived and maintained exclusively within the protected memory of the TEE, it remains inaccessible to any adversary outside the enclave. Consequently, the attestation mechanism ensures that the established communication channel is cryptographically bound to a specific, verified enclave instance.

G2: TEE Mutual Authentication. GroupTEE achieves mutual authentication by combining CA-certified identities with the runtime attestation mechanism described in *G1*. Specifically, the CA binds each participant's identity to its public key and platform measurement. During the protocol, peers verify these credentials together with the fresh attestation evidence to validate both the counterparty's identity and its execution environment.

G3: Session-Key Confidentiality and Consistency. We provide a formal security analysis in the ROM [2], proving that GroupTEE achieves EUF-CMA authentication [6] and session-key indistinguishability. We further prove key consistency via a formula derivation, showing that all honest participants under the same authenticated session context derive the same group session key GK. Given the derived traffic key TK, channel data confidentiality and integrity follow from the security of the underlying AEAD scheme. To eliminate potential logic-level flaws, we complement the computational proof with BAN logic [4] for belief reasoning and symbolic verification using the Scyther tool under the Dolev–Yao adversary model. Detailed proofs of *G3* are provided in our technical report [7].

G4: Session Independence and Freshness. GroupTEE ensures replay resistance and session independence by cryptographically binding each protocol message to a unique session identifier *sid* via signatures and hash inputs. Moreover, the group session key GK is derived from fresh TEE-protected ephemeral randomness and bound to the authenticated session context, ensuring that compromise of a single traffic key TK does not affect other sessions. This design provides standard forward secrecy and session-key independence across protocol executions.

6 Performance Evaluation

6.1 Communication Overhead

We evaluate the communication overhead of GroupTEE in comparison with TLS+RA and MultiTEE. The comparison considers (i) the number of logical rounds, (ii) per-participant message complexity, and (iii) per-message size under two representative topologies, namely fully connected group communication and one-to-many broadcast dissemination. The results are summarized in Table 2.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17

Table 2: Comparison of Communication Overheads.

Scheme	Rounds	Messages per P_i		Per-Message Size
		Broadcast	Group Communication	
TLS+RA	$O(1)$ per link	Sender: $O(n)$ Receiver: $O(1)$	$O(n)$	$O(1)$
MultiTEE	$O(1)$ per link	Sender: $O(n)$ Receiver: $O(1)$	$O(n)$	$O(1)$
GroupTEE	$O(1)$	Sender: $O(1)$ Receiver: $O(1)$	Sender: $O(1)$ Receiver: $O(n)$	$O(1)$

In the fully connected setting, TLS+RA and MultiTEE require each participant to establish $n-1$ independent pairwise secure channels, resulting in $O(n)$ messages per participant. In contrast, GroupTEE completes key establishment within a single group-oriented execution. Each participant transmits one constant-size broadcast message and receives $n-1$ messages, preserving $O(n)$ reception cost while eliminating redundant channel setups.

In the broadcast setting, the advantage of GroupTEE becomes pronounced. TLS+RA and MultiTEE still incur $O(n)$ sender-side communication overhead due to repeated pairwise channel establishment. By contrast, GroupTEE enables a single broadcast dissemination under a shared group key, reducing sender-side communication complexity to $O(1)$ with constant per-message size.

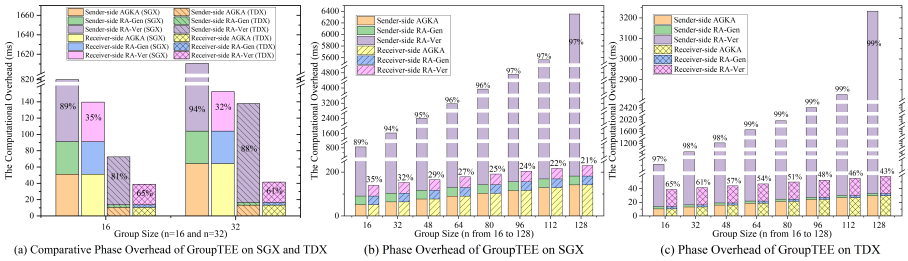
6.2 Computational Overhead

We evaluate GroupTEE under both SGX and TDX environments to quantitatively analyze its computational overhead across different TEE implementations. For SGX, experiments are conducted on an Alibaba Cloud ECS c7t.large instance equipped with an Intel Xeon Platinum 8369B CPU (2.70GHz) with SGX enabled. For TDX, we use an Alibaba Cloud ECS g8i.xlarge instance with Intel TDX enabled.

In our evaluation, we quantify the computational overhead in terms of total execution time, decomposed into three critical phases: authenticated group key agreement (AGKA), remote attestation generation (RA-Gen), and remote attestation verification (RA-Ver). SM2 is adopted for elliptic-curve operations and SM3 for hashing and key derivation. All cryptographic primitives are implemented using the MIRACL Core library [5]. To quantify the computational overhead, we measure the execution time of individual cryptographic primitives and TEE-specific operations in isolation. The average results are reported in Table 3, while the aggregated protocol-level overhead is shown in Table 4.

Fig. 5 illustrates the phase-wise computational overhead under the broadcast topology for both SGX and TDX. As shown in Fig. 5 (a), for fixed group sizes ($n = 16$ and $n = 32$), RA verification dominates the total execution time, accounting for the majority of the overhead. Fig. 5 (b) and (c) further demonstrate

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17



(a) Comparative Phase Overhead of GroupTEE on SGX and TDX (b) Phase Overhead of GroupTEE on SGX (c) Phase Overhead of GroupTEE on TDX

Fig. 5: Computational overhead in broadcast topology

Table 3: Computation time of Cryptographic Primitives.

Operation	Environment		Description	Time/ms
T_h^{app}	SGX	Application	Hash operation	0.0055
T_{ecm}^{app}		Application	Elliptic curve scalar multiplication	1.24
T_h^{enc}		Enclave	Hash operation	0.0056
T_{ecm}^{enc}		Enclave	Elliptic curve scalar multiplication	1.54
$T_{\text{RA-gen}}$		-	Generate RA	39.96
$T_{\text{RA-ver}}$		-	Verify RA	48.59
T_h	TDX	-	Hash operation	0.0053
T_{ecm}		-	Elliptic curve scalar multiplication	0.28
$T_{\text{RA-gen}}$		-	Generate RA	3.22
$T_{\text{RA-ver}}$		-	Verify RA	25.2

that, as the group size increases from 16 to 128, the total cost grows approximately linearly with n. This linear growth is primarily attributed to RA verification, whose cost scales proportionally with the number of peers. In contrast, AGKA and RA generation incur relatively smaller and more stable overhead. Across all group sizes, TDX consistently achieves lower latency than SGX while preserving the same linear scalability pattern.

We consider the worst-case fully connected setting, where each participant P_i must mutually authenticate and establish trust with all other participants. Under this model, we count the number of pairwise key-agreement executions, RA-Gen and RA-Ver operations, and certificate-verification computations performed by P_i in TLS+RA, MultiTEE, and GroupTEE. Table 5 summarizes the resulting per-participant computational overhead.

Both TLS+RA and MultiTEE rely on pairwise channel establishment, requiring each participant to execute $n-1$ independent key agreements and to perform RA generation and verification per peer. In contrast, GroupTEE establishes group-level security semantics and derives a shared group key via a single group-oriented protocol execution per session. Consequently, each participant generates attestation evidence once per session and verifies the other $n-1$ participants' evidence. Since RA generation and verification dominate the computation in our setting, GroupTEE substantially reduces the dominant per-participant computation cost. Moreover, GroupTEE provides native support for

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17

Table 4: Computational overhead for P_i.

Phase		Broadcast		Group Communication
		S-to-R	Mutual Attestation	
Channel Establishment	SGX	$(n+6)T_{ecm}^{enc} + (n+4)T_{h}^{enc} + (2n-2)T_{ecm}^{app}$		
	TDX	$(3n+4)T_{ecm\text{-}e} + (n+4)T_{h\text{-}e}$		
Trusted Attestation	Sender	$(n-1)T_{\text{RA-ver}}$	$T_{\text{RA-gen}} + (n-1)T_{\text{RA-ver}}$	$T_{\text{RA-gen}}+$ $(n-1)T_{\text{RA-ver}}$
	Receiver	$T_{\text{RA-gen}}$	$T_{\text{RA-gen}} + T_{\text{RA-ver}}$	

Table 5: Comparison of Computational Overheads for P_i.

Operation at P_i	TLS+RA	MultiTEE	GroupTEE
Pairwise key agreements	$n-1$	$n-1$	1
Scalar multiplications	$O(n)$	$O(n)$	$O(n)$
RA generations	$n-1$	$n-1$	1
RA verifications	$n-1$	$n-1$	$n-1$
Certificate verifications	$n-1$	0	$n-1$
Group key derivation	×	×	✓

group key establishment, which is not provided by the pairwise TLS+RA and MultiTEE.

7 Conclusion

We present GroupTEE, a group-oriented secure communication framework for mutually distrustful TEEs. By integrating authenticated group key agreement with attestation-based trust binding, GroupTEE enables group-level trust semantics within a single protocol execution. Our analysis shows that GroupTEE achieves constant-round communication and reduces attestation generation from $O(n)$ to $O(1)$ per participant. Experimental results on SGX and TDX demonstrate linear scalability and practical efficiency. Although attestation verification remains $O(n)$ in fully connected settings, GroupTEE provides a scalable foundation for enclave-to-enclave secure group communication. In future work, we will investigate hierarchical designs to further reduce overhead in large-scale deployments.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (U24A20240, 62441226, 62562017).

References

1. Barnes, R., Beurdouche, B., Robert, R., Millican, J., Omara, E., Cohn-Gordon, K.: Rfc 9420: The messaging layer security (mls) protocol (2023)
2. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security. pp. 62–73 (1993)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17

3. Boos, P., Lacoste, M.: Networks of trusted execution environments for data protection in cooperative vehicular systems. In: Vehicular Ad-hoc Networks for Smart Cities: Third International Workshop, 2019. pp. 99–109. Springer (2020)
4. Burrows, M., Abadi, M., Needham, R.: A logic of authentication. ACM Transactions on Computer Systems (TOCS) **8**(1), 18–36 (1990)
5. Consortium, M.: Miracl core library (2024), `https://github.com/miracl/core`, open-source cryptographic library
6. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on computing **17**(2), 281–308 (1988)
7. IIEJan: Security analysis of grouptee. `https://github.com/IIEJan/GroupTEE` (2026)
8. Kim, Y., Perrig, A., Tsudik, G.: Tree-based group key agreement. ACM Transactions on Information and System Security (TISSEC) **7**(1), 60–96 (2004)
9. King, G., Wang, H.: Httpa: Https attestable protocol. In: Future of Information and Communication Conference. pp. 811–823. Springer (2023)
10. King, G., Wang, H.: Httpa/2: A trusted end-to-end protocol for web services. In: Future of Information and Communication Conference. pp. 824–848. Springer (2023)
11. Majumder, A., Warsi, S., Alam, J., Maity, S.: Algebraic signature based trusted execution environment for remote computation. In: Proceedings of the 26th International Conference on Distributed Computing and Networking. pp. 289–294 (2025)
12. Nir, Y., Langley, A.: Rfc 8439: Chacha20 and poly1305 for ietf protocols (2018)
13. Ott, S., Orthen, B., Weidinger, A., Horsch, J., Nayani, V., Ekberg, J.E.: Multitee: Distributing trusted execution environments. In: Proceedings of the 19th ACM Asia Conference on Computer and Communications Security. pp. 1617–1629 (2024)
14. Popa, R.A.: Confidential computing or cryptographic computing? Communications of the ACM **67**(12), 44–51 (2024)
15. Rescorla, E.: The transport layer security (tls) protocol version 1.3. Tech. rep. (2018)
16. Wallez, T., Protzenko, J., Bhargavan, K.: Treekem: A modular machine-checked symbolic security analysis of group key agreement in messaging layer security. In: 2025 IEEE Symposium on Security and Privacy (SP). pp. 4375–4390. IEEE (2025)
17. Walther, R., Weinhold, C., Roitzsch, M.: Ratls: Integrating transport layer security with remote attestation. In: International Conference on Applied Cryptography and Network Security. pp. 361–379. Springer (2022)
18. Wang, L., Tian, Y., Zhang, D., Lu, Y.: Constant-round authenticated and dynamic group key agreement protocol for d2d group communications. Information Sciences **503**, 61–71 (2019)
19. Weinhold, C., Sardar, M.U., Mihalcea, I., Deshpande, Y., Tschofenig, H., Sheffer, Y., Fossati, T., Roitzsch, M.: Separate but together: Integrating remote attestation into {TLS}. In: 2025 USENIX Annual Technical Conference (USENIX ATC 25). pp. 1319–1326 (2025)
20. Xiong, H., Wu, Y., Lu, Z.: A survey of group key agreement protocols with constant rounds. ACM Computing Surveys (CSUR) **52**(3), 1–32 (2019)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29921-5_17