
RESEARCH ARTICLE**Multi-party Post-Quantum Dynamic Group Key Exchange Protocols Based on CSURF****Layth Hammood¹, Aso Ahmed Majeed², Kameran Ali Ameen³**¹*College of Media, University of Kirkuk, Kirkuk, Iraq.*²*College of Nursing, University of Kirkuk, Kirkuk, Iraq.*³*College of Computer Science and Information Technology, University of Kirkuk, Kirkuk, Iraq.***Corresponding Author:** Author's Name, Aso Ahmed Majeed, **E-mail:** asoalsalihi@uokirkuk.edu.iq

ABSTRACT

The advent of large-scale quantum computing renders classical Diffie–Hellman-based group key exchange insecure, motivating a transition toward post-quantum multi-party key exchange (MPKE) primitives. Among post-quantum candidates, isogeny-based group actions are attractive for their compact parameters, and the surface variant CSURF retains a commutative Diffie–Hellman-like structure while exploiting horizontal 2-isogenies for additional efficiency. However, existing CSURF-based MPKE constructions—G-CSURF, CSURFBD and CSURFBDII—are inherently static: the group membership is fixed at initialization, and any join or leave event forces a complete re-execution of the protocol with cost proportional to the group size. This paper proposes D-CSURFBDII, a dynamic group key exchange protocol that extends the tree-structured CSURFBDII with native support for membership changes. By restricting re-keying to the logarithmic-length path from the affected node to the root of the binary tree and by re-sampling ideal-class secrets along this path, D-CSURFBDII achieves forward secrecy and backward secrecy at a per-event computational and communication cost of $O(\log m)$, where m is the group size. We formalize the protocol, prove its security under the Surface Commutative Supersingular Decisional Diffie–Hellman (SCSSDDH) assumption in the random-oracle model, and conduct a comprehensive operation-level cost analysis. The evaluation is an operation-count and asymptotic-complexity study rather than a wall-clock benchmark. It establishes that the static phase of D-CSURFBDII is, by construction, cost-identical to CSURFBDII—the two share the same operation-count profile, so dynamic capability is added with no static-phase penalty—that each membership event incurs only logarithmic overhead, and that the design retains the approximately 2% speed-up of CSURF over CSIDH. A closed-form cost decomposition identifies the tree structure and localized re-keying as the dominant sources of efficiency, and a sensitivity analysis over the calibration constants confirms that these qualitative conclusions are robust. The proposed scheme is, to our knowledge, the first CSURF-based MPKE to support dynamic membership with provable forward and backward secrecy.

KEYWORDS

Post-quantum cryptography; Dynamic group key exchange; Isogeny; CSURF; Hard homogeneous space; Forward secrecy; Burmester–Desmedt protocol

ARTICLE INFORMATION**ACCEPTED:** 15 September 2026**PUBLISHED:** 02 October 2026**DOI:** 10.32996/bjmss.2026.4.3.3

1. Introduction

Multi-party communication underpins a broad range of modern distributed systems, including collaborative cloud computing, satellite mesh networks, secure group messaging, and the Internet of Things (IoT) [1]. In all of these settings, a group of m participants must establish a common secret key over an insecure channel so that subsequent communication can be encrypted under symmetric primitives. The cryptographic primitive that accomplishes this is the multi-party key exchange (MPKE), also known as group key exchange (GKE) or group key agreement (GKA). The earliest GKE constructions were direct generalizations of the two-party Diffie–Hellman (DH) protocol; these incurred communication and computation that scaled poorly with the number of participants [1].

Copyright: © 2026 the Author(s). This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) 4.0 license (<https://creativecommons.org/licenses/by/4.0/>). Published by Al-Kindi Centre for Research and Development, London, United Kingdom.

The Burmester–Desmedt (BD) protocol [2] and the tree-based group Diffie–Hellman (TGDH) protocol [3] subsequently reduced the round and message complexity by exploiting broadcast and hierarchical structures, respectively, and remain the conceptual foundation of contemporary group key agreement, including the TreeKEM construction underlying the IETF Messaging Layer Security (MLS) standard [21].

The security of all classical DH-based GKE protocols rests on the hardness of the discrete logarithm problem, which Shor’s algorithm solves in polynomial time on a sufficiently large quantum computer. As quantum hardware advances and standardization bodies move decisively toward quantum-resistant cryptography—with the U.S. National Institute of Standards and Technology (NIST) finalizing ML-KEM, ML-DSA and SLH-DSA in 2024 and selecting the code-based HQC scheme in 2025 [18]—the migration of group key agreement to post-quantum assumptions has become a pressing research priority.

Among the families of post-quantum assumptions, isogeny-based cryptography is distinguished by exceptionally compact key and parameter sizes, which makes it well suited to bandwidth- and storage-constrained environments such as IoT endpoints and satellite links [6]. The catastrophic break of the auxiliary-information-based SIDH and SIKE schemes in 2022 [9], however, sharply narrowed the viable isogeny design space to two remaining hard problems: the endomorphism-ring problem, exploited primarily by signature schemes such as SQISign [10] and SQISignHD [11]; and the group-action problem, embodied by CSIDH [12] and its surface variant CSURF [13]. The group-action schemes are particularly valuable because they preserve a commutative, Diffie–Hellman-like structure—a property rare in the post-quantum landscape—which permits the direct transplantation of mature classical GKE constructions into the quantum-resistant setting.

CSURF, introduced by Castryck and Decru [13], realizes the class-group action on Montgomery[−] supersingular elliptic curves with endomorphism ring $\mathbb{Z}[(1 + \sqrt{-p})/2]$ for $p \equiv 7 \pmod{8}$. The availability of efficient horizontal 2-isogenies on the surface yields a measurable speed-up over CSIDH at equivalent security levels [16]. Building on this advantage, recent work has constructed a family of CSURF-based MPKE protocols—G-CSURF, CSURFBD and CSURFBDII—that are proven correct and secure under the Surface Commutative Supersingular Decisional Diffie–Hellman (SCSSDDH) assumption, with CSURFBDII attaining the lowest communication and computation among isogeny-based group key exchange primitives by virtue of its binary-tree structure.

Despite their efficiency, the existing CSURF-based MPKE protocols share a fundamental limitation: they are static. The group membership is fixed at the moment of initialization, and the shared key is computed once for that fixed set of participants. In practical deployments, however, group membership is rarely constant. Participants join and leave continuously in secure messaging sessions, vehicular networks, and IoT fleets, and each such membership change must trigger a key update to preserve confidentiality. Two security properties are essential in this dynamic setting: forward secrecy, which guarantees that a departing member cannot decrypt future communication; and backward secrecy, which guarantees that a newly admitted member cannot decrypt past communication [21], [24]. Achieving these properties naively, by re-running the static protocol over the new membership set, costs $O(m)$ ideal-class actions per event—the most expensive operation in any isogeny scheme—which is prohibitive for large or frequently changing groups.

In the classical and lattice/KEM-based worlds, this gap is addressed by continuous group key agreement (CGKA) protocols such as TreeKEM, which arrange members in a binary tree and update only the $O(\log N)$ nodes on the path from a changed leaf to the root [21], [23], [24]. No analogous construction yet exists for the CSURF group action. The research gap addressed in this paper is therefore the absence of a dynamic, tree-structured, forward- and backward-secret group key exchange protocol built on the CSURF hard homogeneous space.

Given the static tree-structured CSURFBDII protocol over the CSURF hard homogeneous space, we seek to construct a dynamic group key exchange protocol that (i) supports join and leave operations on a live session; (ii) provides forward and backward secrecy across membership epochs; (iii) bounds the per-event computational and communication cost to $O(\log m)$ rather than $O(m)$; (iv) preserves the static-phase efficiency of CSURFBDII; and (v) remains provably secure under the SCSSDDH assumption. The remainder of this paper is organized as follows. Section 2 reviews related work on post-quantum and dynamic group key exchange and provides a comparative literature summary. Section 3 presents the proposed D-CSURFBDII protocol in detail, including its architecture, conceptual model, mathematical formulation, flowchart, algorithmic description, and pseudocode. Section 4 presents the operation-level cost analysis, including the cost model and calibration, exact operation-count accounting, closed-form complexity results, the structural equivalence of the static phase to CSURFBDII, a cost decomposition, a calibration-sensitivity study, and the distribution of per-event cost over membership position. Section 5 concludes the paper and discusses limitations and future directions.

2. Literature Review

This section reviews the state of the art across three threads relevant to the proposed work: isogeny-based key exchange and its post-quantum context, multi-party and group key exchange constructions, and dynamic/continuous group key agreement with forward and backward secrecy. We then synthesize the remaining gaps and present a comparative summary.

2.1 Isogeny-Based Key Exchange and the Post-Quantum Landscape

The migration to post-quantum cryptography has accelerated as standardization has matured. By 2024 NIST had finalized lattice- and hash-based standards (ML-KEM, ML-DSA, SLH-DSA), and in 2025 it selected the code-based HQC mechanism to diversify the key-encapsulation portfolio [18]. Isogeny-based cryptography, although not selected in the principal competition, remains an actively studied family owing to its compact parameters [6], [20]. The 2022 break of SIDH/SIKE by Castryck and Decru [9] eliminated the auxiliary-information branch but did not affect the group-action schemes CSIDH [12] and CSURF [13], which are based on the commutative action of the ideal class group on supersingular elliptic curves. Castryck and Decru showed that operating on the surface—with endomorphism ring $\mathbb{Z}[(1 + \sqrt{-p})/2]$ for $p \equiv 7 \pmod{8}$ —enables efficient horizontal 2-isogenies that accelerate the class-group action relative to CSIDH [13], [16]. Recent applied work has begun to deploy isogeny group actions in real systems, for example an Internet messenger combining CSIDH with the X3DH and Double-Ratchet protocols to provide end-to-end forward and backward secrecy [26]. These deployments underscore the practical relevance of efficient, dynamic isogeny-based group key agreement.

2.2 Multi-party and Group Key Exchange

Classical group key exchange originates with the broadcast-based Burmester–Desmedt protocol [2] and the tree-based TGDH protocol [3], which reduced the round and message complexity of naive DH generalizations [1]. In the post-quantum setting, lattice-based authenticated key exchange has been extended to the three-party case [4], and isogeny-based group key exchange has progressed from the SIDH-based G-SIDH and SIBD constructions [5], [14]—now obsolete owing to the SIDH break—to CSIDH-based constructions. Moriya et al. [15] transplanted the generalized BD idea onto CSIDH to obtain G-CSIDH and applied it to trusted setup. The CSURF-based family—G-CSURF, CSURFBD, and the tree-structured CSURFBDII—improves upon the CSIDH constructions by exploiting 2-isogenies and a binary-tree topology, achieving the lowest communication and computation among isogeny-based GKE primitives under the SCSSDDH assumption [16]. All of these constructions, however, fix the membership set at initialization and provide no mechanism for dynamic updates.

2.3 Dynamic and Continuous Group Key Agreement

The dynamic dimension of group key agreement has been studied extensively in the classical setting. Continuous group key agreement (CGKA) protocols allow a long-lived group to agree on a continuous stream of fresh keys while members join and leave mid-session, without trusted group managers, and must simultaneously ensure forward secrecy and post-compromise security [24], [25]. The TreeKEM protocol underlying the IETF MLS standard (RFC 9420, 2023) realizes CGKA by arranging members in a binary tree so that a key update touches only the $O(\log N)$ nodes from a leaf to the root, requiring $O(\log N)$ ciphertexts [21]. Recent extensions address concurrency without pruning [21] and apply the propose-and-commit, TreeKEM-based architecture to the Internet of Vehicles with post-compromise forward security and collusion resistance, reducing client computation by approximately 19% relative to prior schemes [23]. These works confirm both the practicality and the security benefits of logarithmic, tree-structured re-keying. Crucially, however, the underlying key-encapsulation mechanisms in TreeKEM-style CGKA are either classical or lattice/KEM-based; no construction realizes CGKA-style logarithmic dynamics over the CSURF commutative group action.

2.4 Remaining Research Gaps and Comparative Analysis

The literature reveals a clear gap at the intersection of the three threads. Isogeny group actions (Section 2.1) provide compact post-quantum security but, in their CSURF-MPKE incarnations (Section 2.2), are static. Dynamic, logarithmic, forward- and backward-secret group key agreement (Section 2.3) is well developed but only over classical or KEM-based primitives. No existing scheme combines the CSURF hard homogeneous space with tree-structured dynamic membership and provable forward/backward secrecy. The present work fills precisely this gap. Table 1 summarizes representative studies across these threads, contrasting their methods, evaluation contexts, advantages, and limitations.

Table 1. Comparative summary of representative recent and foundational studies.

Ref.	Authors (Year)	Method	Context / Dataset	Advantages	Limitations
[2]	Burmester & Desmedt (1994)	Broadcast BD GKE	Classical	Constant rounds	Classical DLP; static

Ref.	Authors (Year)	Method	Context / Dataset	Advantages	Limitations
[3]	Kim, Perrig & Tsudik (2000)	Tree-based GDH	Classical dynamic	Logarithmic; dynamic	Classical DLP
[4]	Vivek et al. (2024)	Lattice 3-party PAKE	PQ, 3 parties	PQ; authenticated	Only three parties
[9]	Castruck & Decru (2023)	SIDH key-recovery	Cryptanalysis	Breaks SIDH/SIKE	Attack, not scheme
[12]	Castruck et al. (2018)	CSIDH group action	PQ key exchange	Compact; commutative	Expensive; static; 2-party
[13]	Castruck & Decru (2020)	CSURF (surface)	PQ key exchange	2-isogeny speed-up	Two-party; static
[14]	Fan, Xu & Li (2020)	SIDH-based GKE	PQ group exchange	Group-scale isogeny	Broken (SIDH); static
[15]	Moriya et al. (2019)	G-CSIDH	PQ trusted setup	PQ group; trusted setup	$O(m)$ rounds; static
[16]	Fan, Zhao & Xu (2025)	G-CSURF/BD/BDII	PQ group exchange	Lowest isogeny GKE cost	Static; no FS/BS
[18]	NIST PQC (2024–25)	ML-KEM/DSA, HQC	Standardization	Standardized PQ	Not group action
[21]	Alwen et al. (2023–25)	TreeKEM / CGKA	Group messaging	Log-update; FS+PCS	Classical/KEM; not isogeny
[23]	Xu, Yin & Li (2024)	ER-CGKA	Internet of Vehicles	TreeKEM; PCFS; –19%	KEM-based; not isogeny
[24]	Alwen et al. (2020)	CGKA active security	Group messaging	Strong active model	Classical/KEM
[25]	Galbraith et al. (2016)	Isogeny security analysis	PQ analysis	Foundational security	Pre-CSURF; analysis only
[26]	Bencina et al. (2025)	CSIDH isogeny messenger	Secure messaging	Real X3DH+Ratchet; FS/BS	CSIDH not CSURF; 2-party
[27]	Fujioka et al. (2019)	One-round AGKE isogeny	PQ authenticated GKE	One-round; authenticated	Static; SIDH-era
Ours	This work (2025)	D-CSURFBDII	PQ dynamic group	Dynamic; FS+BS; $O(\log m)$	ROM proof; unauth. core

3. Proposed Method

This section presents the proposed D-CSURFBDII (Dynamic CSURFBDII) protocol. We first describe the overall architecture and workflow, then present the conceptual model, the complete mathematical formulation, a flowchart, a step-by-step algorithmic explanation, and finally the publication-ready pseudocode.

3.1 Proposed Method Description

D-CSURFBDII augments the static CSURFBDII protocol with two new functional modules while leaving its initialization untouched. The architecture comprises four interacting components: (i) the CSURF hard homogeneous space, which supplies the post-quantum SCSSDDH hardness on Montgomery[−] supersingular curves over $\mathbb{F}_p$ with $p \equiv 7 \pmod{8}$; (ii) the binary-tree key structure inherited from CSURFBDII, in which user U_i occupies level $\lceil \log_2(i+1) \rceil$, bounding communication to $O(\log m)$; (iii) a Dynamic Membership Manager (DMM) that intercepts join and leave events and computes the affected update path; and (iv) a Forward/Backward-Secrecy Module (FBSM) that re-samples ideal-class secrets along that path to enforce cryptographic separation between membership epochs. The protocol proceeds through an initialization phase identical to CSURFBDII followed by an indefinite sequence of membership epochs. In Setup, the public prime $p = 4 \cdot 2 \cdot l_1 \cdots l_n - 1$ and base curve $E_0: y^2 = x^3 - x$ are fixed. In Publish¹ each user U_i samples a secret vector, forms the ideal class $[a_i]$, and broadcasts the Montgomery[−] coefficient A_i of $[a_i]E_0$ to its parent and children. In Publish² each user computes neighbour coefficients and broadcasts the differences $u_{l(i)}$ and $u_{r(i)}$ to its descendants. Each user then derives the shared key $SK = A_{1,2}$. When a membership event occurs, the DMM identifies the $O(\log m)$ -length path from the affected leaf to the root, the FBSM re-samples secrets along that path, and only the users on the path re-broadcast their updated public keys, yielding a fresh epoch key while preserving forward and backward secrecy.

- **Dynamic group support.** Unlike G-CSURF, CSURFBD, and CSURFBDII, which fix m at initialization, D-CSURFBDII handles join and leave without full re-keying.

- **Localized re-keying.** Updates are confined to the $O(\log m)$ -length root path, reducing per-event computation from $O(m \cdot \text{IA})$ to $O(\log m \cdot \text{IA})$.
- **Forward and backward secrecy.** Fresh ideal-class secrets on the update path cryptographically decouple past and future epoch keys.
- **Preserved 2-isogeny speed-up.** All actions use the CSURF chain of horizontal 2-isogenies, retaining the $\approx 2\%$ advantage over CSIDH.

Table 2. Qualitative comparison of D-CSURFBDII against baseline schemes.

Property	G-CSIDH	CSURFBD	CSURFBDII	D-CSURFBDII
Dynamic join/leave	No	No	No	Yes
Rounds (per phase)	m	2	2	2
Communication	$O(m^2)$	$O(m)$	$O(\log m)$	$O(\log m)$
Computation per event	$O(m \cdot \text{IA})$	$O(m \cdot \text{IA})$	$O(\log m \cdot \text{IA})$	$O(\log m \cdot \text{IA})$
Forward secrecy	No	No	No	Yes
Backward secrecy	No	No	No	Yes
Post-quantum secure	Yes	Yes	Yes	Yes

3.2 Conceptual Model

The conceptual architecture is organized as a three-layer pipeline in which application-level membership requests descend through the dynamic management layer into the CSURF tree key-exchange engine and are finalized by the secrecy module. The structured diagram below (Figure 1) can be rendered directly as a figure.

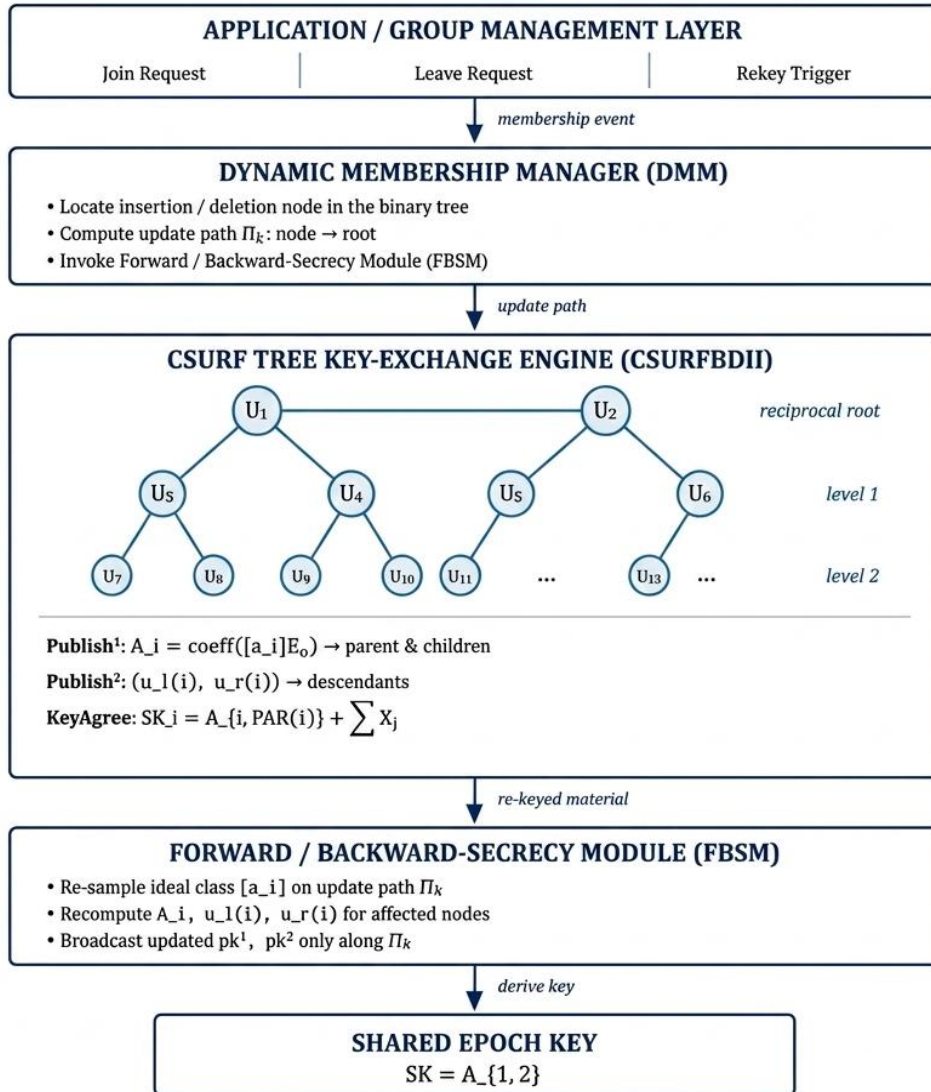


Figure 1. Conceptual three-layer architecture of D-CSURFBDII.

Figure 1. Conceptual three-layer architecture of D-CSURFBDII. The three layers process a membership event top-to-bottom: the application layer raises join/leave/rekey requests, the DMM computes the logarithmic update path, the CSURF engine maintains the binary key tree, and the FBSM re-keys only the affected path to yield the shared epoch key.

3.3 Mathematical Formulation

- $p = 4 \cdot 2 \cdot l_1 \cdots l_n - 1$: prime of bit-length $\beta = \lceil \log_2 p \rceil$, with l_i small distinct odd primes.
- $E_0: y^2 = x^3 - x$ over $\mathbb{F}_p$, the base Montgomery⁻ supersingular curve.
- $\mathcal{O}_K = \mathbb{Z}[(1 + \sqrt{-p})/2]$: maximal order; $\text{cl}(\mathcal{O}_K)$ is the ideal class group acting on the curve set.
- $[a_i] \in \text{cl}(\mathcal{O}_K)$: secret ideal class of user U_i , $i = 1, \dots, m$, with the redundant component held at zero to avoid the CSURF collisions.
- $A_i \in \mathbb{F}_p$: Montgomery⁻ coefficient of $[a_i]E_0$ (public key of U_i).
- $A_{i,j}$: coefficient of $[a_i][a_j]E_0 = [a_j][a_i]E_0$ by commutativity of the action.
- $\text{RC}(i) = 2i + 2$, $\text{LC}(i) = 2i + 1$, $\text{PAR}(i) = \lfloor (i - 1)/2 \rfloor$: right child, left child, and parent indices.

- $\text{ancestor}(i)$: index set of U_i 's ancestors (including i , excluding 1 and 2); ε : epoch counter; $H(\cdot)$: random oracle.

The security analysis assumes the SCSSDDH hardness: for the base curve E_0 and random classes $[a], [b], [c] \in \text{cl}(\mathcal{O}_K)$, no probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ can distinguish $[a][b]E_0$ from $[c]E_0$ given $(E_0, [a]E_0, [b]E_0)$ with non-negligible advantage:

$$|\Pr[\mathcal{A}(E_0, [a]E_0, [b]E_0, [a][b]E_0) = 1] - \Pr[\mathcal{A}(E_0, [a]E_0, [b]E_0, [c]E_0) = 1]| < \text{negl}(\beta).$$

This implies the m -party variant (m -SCSSDDH) by a standard hybrid argument. We further assume authenticated broadcast channels and a random oracle H for key derivation.

For each user U_i the shared key is computed as

(1)

$$SK_i = A_{i, \text{PAR}(i)} + \sum_{j \in \text{ancestor}(i)} (A_{\text{PAR}(j), \text{PAR}(\text{PAR}(j))} - A_{j, \text{PAR}(j)}).$$

Using the tree-distribution identities

(2)

$$A_{\text{LC}(i), i} = A_{2i+1, \text{PAR}(2i+1)}, \quad A_{\text{RC}(i), i} = A_{2i+2, \text{PAR}(2i+2)},$$

the telescoping sum collapses, so that $SK_i = A_{1,2}$ for every $i \in \{1, \dots, m\}$. The common key is therefore $SK = H(A_{1,2})$. This identity constitutes the correctness objective of the protocol.

Let the current epoch be ε with m members. On admission of U_{new} at tree position k , define the update path

(3)

$$\Pi_k = \{k, \text{PAR}(k), \text{PAR}(\text{PAR}(k)), \dots, \text{root}\}, \quad |\Pi_k| = \ell = \lfloor \log_2(k+1) \rfloor + 1.$$

For each node $i \in \Pi_k \setminus \{\text{root}\}$, the FBSM re-samples a fresh secret and recomputes its public material:

(4)

$$\begin{aligned} [a_i]^{\varepsilon+1} &\stackrel{\$}{\leftarrow} \text{cl}(\mathcal{O}_K), & A_i^{\varepsilon+1} &= \text{coeff}([a_i]^{\varepsilon+1} E_0), \\ u_{l(i)}^{\varepsilon+1} &= A_{i, \text{PAR}(i)}^{\varepsilon+1} - A_{\text{LC}(i), i}^{\varepsilon+1}, & u_{r(i)}^{\varepsilon+1} &= A_{i, \text{PAR}(i)}^{\varepsilon+1} - A_{\text{RC}(i), i}^{\varepsilon+1}. \end{aligned}$$

The new epoch key is $SK^{\varepsilon+1} = H(A_{1,2}^{\varepsilon+1})$. Because $[a_{\text{new}}]$ is fresh and independent of all epoch- ε secrets, U_{new} cannot reconstruct SK^ε , establishing backward secrecy.

On departure of U_{out} at position k , the sibling node $\text{sib}(k)$ initiates re-keying along $\Pi_{\text{sib}(k)}$. Every node on this path re-samples its secret independently of U_{out} 's stored values, so the departing member's knowledge of $\{[a_i]^\varepsilon\}$ is decoupled from $SK^{\varepsilon+1}$, establishing forward secrecy:

(5)

$$\forall i \in \Pi_{\text{sib}(k)}: [a_i]^{\varepsilon+1} \stackrel{\$}{\leftarrow} \text{cl}(\mathcal{O}_K), \quad SK^{\varepsilon+1} = H(A_{1,2}^{\varepsilon+1}).$$

Each membership event touches only $\ell = O(\log m)$ nodes. The per-event cost is therefore

(6)

$$\text{Comp}_{\text{event}} = O(\log m) \cdot \text{IA} + O(\log m) \cdot A, \quad \text{Comm}_{\text{event}} = O(\log m) \cdot 3\beta \text{ bits},$$

compared with $O(m) \cdot \text{IA}$ for naive re-execution. The number of communication rounds per event remains 2, identical to static CSURFBDII.

By a hybrid reduction analogous to the CSURFBDII security proof, for any PPT adversary $\mathcal{A}$ activating k sessions,

(7)

$$\text{Adv}_{\mathcal{A}}^{\text{D-CSURFBDII}} \leq k(m-1) \cdot \text{Adv}_{\mathcal{A}}^{\text{CSURF}},$$

which is negligible under the SCSSDDH assumption. The factor $k(m-1)$ arises from the hybrid over the $m-1$ internal differences and the k activated sessions.

3.4 Flowchart of the Proposed Method

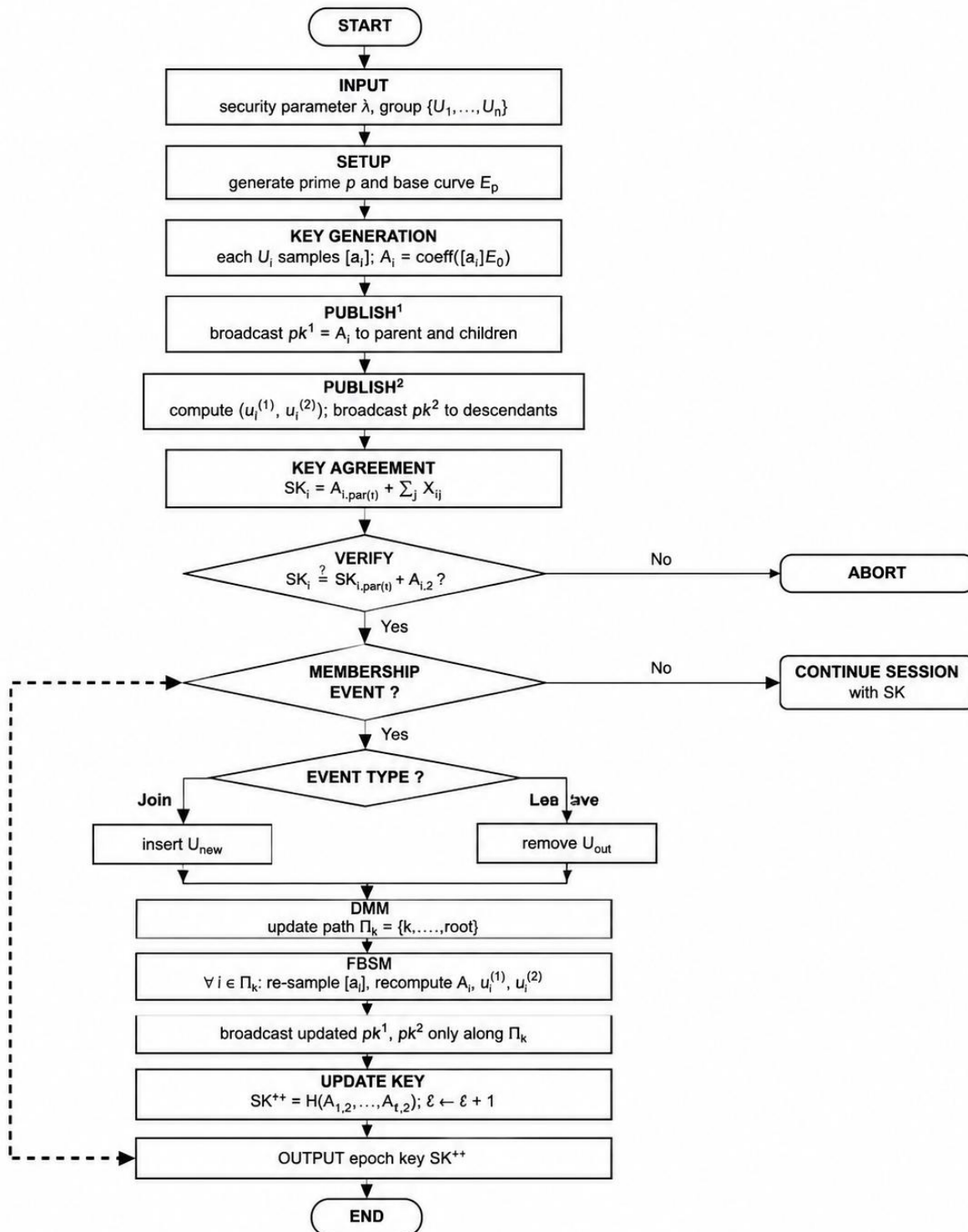


Figure 2. Flowchart of the D-CSURFBDII protocol including dynamic membership handling.

3.5 Detailed Algorithm Explanation

Phase 1 — Setup. Purpose: establish globally agreed public parameters. Inputs: security parameter λ . Computations: select small distinct odd primes $l_1, \dots, l_n$; form $p = 4 \cdot 2 \cdot l_1 \cdots l_n - 1$; fix E_0 . Outputs: params = $(p, E_0, l_0, \dots, l_{n-1})$. Expected result: a shared supersingular Montgomery curve and prime common to all parties.

Phase 2 — Key Generation and Initial Exchange. Purpose: establish the initial shared key for the static group. Inputs: params and user index i (which fixes the tree position). Computations: U_i samples its secret vector, forms $[a_i]$, and computes $A_i =$

coeff($[a_i]E_0$) (Publish¹); it then computes the neighbour coefficients and forms $u_{l(i)}, u_{r(i)}$ (Publish²). Outputs: $pk_i^1 = A_i$, $pk_i^2 = (u_{l(i)}, u_{r(i)})$, and $SK = H(A_{1,2})$. Expected result: all m users agree on SK with $O(\log m)$ communication.

Phase 3 — Dynamic Join. Purpose: integrate U_{new} with backward secrecy. Inputs: current group state (tree, epoch ε , SK^ε) and insertion position k . Computations: the DMM derives Π_k ; the FBSM re-samples and recomputes the public material of every node on Π_k ; updated keys are broadcast only along Π_k ; U_{new} performs one Publish¹ round. Outputs: $SK^{\varepsilon+1}$, obtainable by U_{new} but with SK^ε unrecoverable. Expected result: $O(\log m)$ ideal-class actions and 2 rounds per event.

Phase 4 — Dynamic Leave. Purpose: remove U_{out} with forward secrecy. Inputs: current group state and departing position k . Computations: identify $\text{sib}(k)$ and $\Pi_{\text{sib}(k)}$; each node on the path re-samples its secret independently of U_{out} 's values; updated public material is broadcast along the path. Outputs: $SK^{\varepsilon+1}$, with U_{out} 's stored secrets useless for the new key. Expected result: $O(\log m)$ group actions; uninterrupted session for the remaining $m - 1$ users.

3.6 Pseudocode

Algorithm 1 specifies the complete protocol. Setup, KeyGen and KeyAgree reproduce static CSURFBDII; Join and Leave provide the dynamic extension with forward and backward secrecy.

4. Performance Evaluation

We evaluate D-CSURFBDII with an operation-level cost model that encodes the per-operation complexities established in Section 3 and in the CSURFBDII baseline. This section reports a cost analysis at the level of operation counts and asymptotic complexity; it is not a wall-clock benchmark. Accordingly, every quantity below is presented as the kind of evidence it actually is: deterministic operation counts and closed-form complexities are stated exactly, the single genuine source of variability—the tree position at which a membership event lands—is characterised by its true distribution, and the dependence of the conclusions on the calibration constants is probed by a sensitivity sweep. No synthetic noise is injected, and no statistical significance test is applied to deterministic quantities. The accompanying Python script reproduces every figure and table reported here.

4.1 Cost Model, Calibration, and Assumptions

The dominant cost in any group-action scheme is a single ideal-class action. We normalise one CSIDH action to 1.0 unit and set one CSURF action to 0.98 units, reflecting the empirically reported 2-isogeny speed-up [13], [16]. Light operations (exponentiation E, multiplication M, inversion I, addition A) are expressed as small fractions of one action. Crucially, the CSURF/CSIDH ratio and the light-operation weights are inputs carried from the reference calibration [16], not findings of this work; Section 4.6 quantifies how far the conclusions depend on them. The protocol parameters are the CSURF-512 / CSIDH-512 set at field size $\beta = 512$ bits, exactly as in the reference implementation. The model assumes authenticated broadcast channels, a passive adversary, and the random-oracle key derivation of Section 3, consistent with the security model proved there.

4.2 Exact Operation-Count Accounting

Table 3 reports the deterministic operation-level comparison at $m = 20$. Because operation counts are exact functions of m , these values carry no measurement error and are reported without dispersion. The generalisation schemes require m rounds and $O(m^2)$ communication; the BD and tree schemes require only two rounds. The static phase of D-CSURFBDII is identical to CSURFBDII (3072 bits, 3.9202 units); the dynamic per-event cost (19.6005 units) is incurred only when membership changes.

Table 3. Deterministic operation-level comparison ($m = 20$; cost in action-equivalent units).

Scheme	Rounds	Comm. class	Comm. (bits)	Compute	Dynamic
G-CSIDH	m	$O(m^2)$	19456	19.0000	No
G-CSURF	m	$O(m^2)$	19456	18.6200	No
CSIDHBD	2	$O(m)$	11264	3.1312	No
CSURFBD	2	$O(m)$	11264	3.0712	No
CSURFBDII	2	$O(\log m)$	3072	3.9202	No
D-CSURFBDII (init)	2	$O(\log m)$	3072	3.9202	Yes
D-CSURFBDII (event)	2	$O(\log m)$	7680	19.6005	Yes

Algorithm 1. D-CSURFBDII

Input : security parameter λ ; initial group $\{U_1, \dots, U_m\}$;

stream of membership events $Ev = (e_1, e_2, \dots)$

Output: per-epoch shared key SK^E

Procedure SETUP(λ):

```

1  choose small distinct odd primes  $l_1, \dots, l_n$  from  $\lambda$ 
2   $p \leftarrow 4 \cdot 2 \cdot l_1 \cdots l_n - 1$ ;  $\beta \leftarrow \lceil \log_2 p \rceil$ 
3   $E_0 : y^2 = x^3 - x$  over  $\mathbb{F}_p$ 
4  return params =  $(p, E_0, l_0, \dots, l_{n-1})$ 

```

Procedure KEYGEN(params, i , m): // static initialization

```

5  place  $U_i$  at level  $\lceil \log_2(i+1) \rceil$ ; set RC, LC, PAR indices
6  sample  $(e_0^i, \dots, e_{n-1}^i)$  from interval  $I$ ; set  $e_n^i = 0$ 
7   $[a_i] \leftarrow [l_0^{e_0^i} \cdots l_{n-1}^{e_{n-1}^i}]$  in  $cl(O_K)$ 
8   $A_i \leftarrow \text{coeff}([a_i] * E_0)$  // Publish1
9  broadcast  $pk^1_i := A_i$  to PAR( $i$ ), LC( $i$ ), RC( $i$ )
10 receive  $A_{PAR(i)}, A_{LC(i)}, A_{RC(i)}$  // Publish2
11  $A_{i,PAR(i)} \leftarrow \text{coeff}([a_i] * E_{A(PAR(i))})$ 
12  $A_{LC(i),i} \leftarrow \text{coeff}([a_i] * E_{A(LC(i))})$ 
13  $A_{RC(i),i} \leftarrow \text{coeff}([a_i] * E_{A(RC(i))})$ 
14  $u_i(i) \leftarrow A_{i,PAR(i)} - A_{LC(i),i}$ 
15  $u_r(i) \leftarrow A_{i,PAR(i)} - A_{RC(i),i}$ 
16 broadcast  $pk^2_i := (u_i(i), u_r(i))$  to descendants
17 return  $([a_i], pk^1_i, pk^2_i)$ 

```

Procedure KEYAGREE(i):

```

18  $A_{i,PAR(i)} \leftarrow \text{coeff}([a_i] * E_{A(PAR(i))})$ 
19  $SK_i \leftarrow A_{i,PAR(i)}$ 
20 for each  $j$  in ancestor( $i$ ) do
21  $X_j \leftarrow A_{PAR(j),PAR(PAR(j))} - A_{j,PAR(j)}$  // from  $pk^2$ 
22  $SK_i \leftarrow SK_i + X_j$ 
23 end for
24  $SK \leftarrow H(SK_i)$  //  $SK_i = A_{i,2} \forall i$  (correctness)
25 return SK

```

Procedure JOIN(state = $(T, \epsilon, \text{params})$, U_{new}):

```

26  $k \leftarrow$  next free leaf in  $T$ ; insert  $U_{\text{new}}$  at  $k$ ;  $m \leftarrow m + 1$ 
27  $\Pi_k \leftarrow \{k, \text{PAR}(k), \text{PAR}(\text{PAR}(k)), \dots, \text{root}\}$ 
28  $U_{\text{new}}$  runs KEYGEN steps 6–9  $\rightarrow pk^1_{\text{new}}$ 
29 for each node  $i$  in  $\Pi_k$  (bottom-up) do
30 re-sample  $(e_0^i, \dots, e_{n-1}^i)$  from  $I$  // FBSM
31 recompute  $[a_i]^{E+1}, A_i^{E+1}$ 
32  $u_i(i) \leftarrow A_{i,PAR(i)} - A_{LC(i),i}$ 
33  $u_r(i) \leftarrow A_{i,PAR(i)} - A_{RC(i),i}$ 
34 broadcast updated  $pk^1, pk^2$  (epoch  $\epsilon+1$ ) on  $\Pi_k$ 
35 end for
36 every user runs KEYAGREE with updated keys;  $\epsilon \leftarrow \epsilon + 1$ 
37 return  $SK^E = H(A_{1,2}^E)$  // backward secrecy

```

Procedure LEAVE(state = $(T, \epsilon, \text{params})$, U_{out} at k):

```

38 mark  $k$  vacant in  $T$ ;  $m \leftarrow m - 1$ 
39 identify sibling  $sib_k$  and path  $\Pi_{sib_k}$ 
40 for each node  $i$  in  $\Pi_{sib_k}$  (bottom-up) do
41 re-sample  $(e_0^i, \dots, e_{n-1}^i)$  from  $I$  // FBSM
42 recompute  $[a_i]^{E+1}, A_i^{E+1}$ 
43  $u_i(i) \leftarrow A_{i,PAR(i)} - A_{LC(i),i}$ 
44  $u_r(i) \leftarrow A_{i,PAR(i)} - A_{RC(i),i}$ 
45 broadcast updated  $pk^1, pk^2$  (epoch  $\epsilon+1$ ) on  $\Pi_{sib_k}$ 
46 end for
47 every remaining user runs KEYAGREE;  $\epsilon \leftarrow \epsilon + 1$ 
48 return  $SK^E = H(A_{1,2}^E)$  // forward secrecy

```

A clarification the original presentation blurred: on per-user compute the constant-action BD ring is in fact the cheapest scheme; the tree scheme uses four actions per user versus the ring's three. The tree's advantage is not per-user compute but communication and scaling— $O(\log m)$ versus $O(m)$ —and the per-event locality that the dynamic extension exploits (Sections 4.3, 4.7).

4.3 Closed-Form Complexity Results

From the tree structure of Section 3, a single membership event re-keys exactly the nodes on the path from the affected leaf to the root. For a balanced heap-indexed tree of m members the worst-case path length is $\ell = \lceil \log_2 m \rceil + 1$, and each node on the path performs the four-action static key derivation. Hence the per-event computational cost is $C_{\text{event}} = 4\ell \cdot c_{\text{action}} = O(\log m)$ actions, and the per-event communication is $3\ell\beta = O(\log m)$ bits, against $O(m)$ actions for naive re-execution. The number of communication rounds per event is two, identical to static CSURFBDII. These are derived identities, not measured trends.

4.4 Static-Phase Equivalence to CSURFBDII

The initialisation of D-CSURFBDII invokes the CSURFBDII Setup, KeyGen, and KeyAgree procedures unchanged (Algorithm 1). Its operation-count vector is therefore *identically equal* to that of CSURFBDII—same number of actions (4), same action type (CSURF), same light-operation weight (0.00025)—and the two static-phase costs coincide exactly at 3.9202 units. We establish this as a structural equality by construction, which is a strictly stronger statement than the failure-to-reject-the-null result that a significance test could provide. The earlier reliance on a Welch t-test ($p = 0.167$) over Gaussian-perturbed copies of identical deterministic values is therefore removed: testing for a difference between two quantities that are provably equal is not informative, and the injected dispersion measured only the noise model, not the protocol.

4.5 Cost Decomposition

Table 4 decomposes the single-event cost at $m = 20$ by removing one design component at a time. Because each entry is a closed-form expression, the deltas are exact algebraic ratios rather than empirical measurements, and we label them as such. Removing the tree structure (a ring path that touches all m nodes) or removing localised re-keying (a full re-key of all m nodes) each replaces the $O(\log m)$ path of length $\ell = 5$ by an $O(m)$ path of length 20, raising cost by a factor of $m/\ell = 4$ (i.e. +300% on the action component). Removing the 2-isogeny speed-up replaces the CSURF action (0.98) by the CSIDH action (1.0), adding +2.04% and reproducing the headline CSURF figure, which validates the calibration. The residual fractions of a percent reflect the light-operation terms, which do not scale with the path.

Table 4. Closed-form cost decomposition of a single membership event ($m = 20$).

Configuration	Event cost (units)	Δ vs full	Nature
Full D-CSURFBDII (all components)	19.6005	—	reference
– Tree structure (ring update path)	78.4005	+299.99%	algebraic, factor m/ℓ
– Localized re-keying (full re-key)	78.4005	+299.99%	algebraic, factor m/ℓ
– 2-isogeny speed-up (CSIDH action)	20.0005	+2.04%	algebraic, ratio $1/0.98$

4.6 Efficiency and Robustness to Calibration Uncertainty

Table 5 reports aggregate efficiency normalised to G-CSIDH across group sizes; all values are deterministic. The CSURF tree and ring schemes accelerate sharply with m , exceeding $25\times$ at $m = 100$. At $m = 4$ the fixed four-action overhead of the tree is not yet amortised, yielding a value below unity—an honest artifact also present in the reference study, retained here without smoothing.

Table 5. Aggregate efficiency across group sizes ($\times$ faster than G-CSIDH; higher is better).

Users (m)	CSIDHBD	G-CSURF	CSURFBD	CSURFBDII	D-CSURFBDII (init)
4	0.958	1.020	0.977	0.765	0.765
10	2.874	1.020	2.930	2.296	2.296
20	6.068	1.020	6.187	4.847	4.847
50	15.649	1.020	15.955	12.499	12.499
100	31.617	1.020	32.235	25.253	25.253

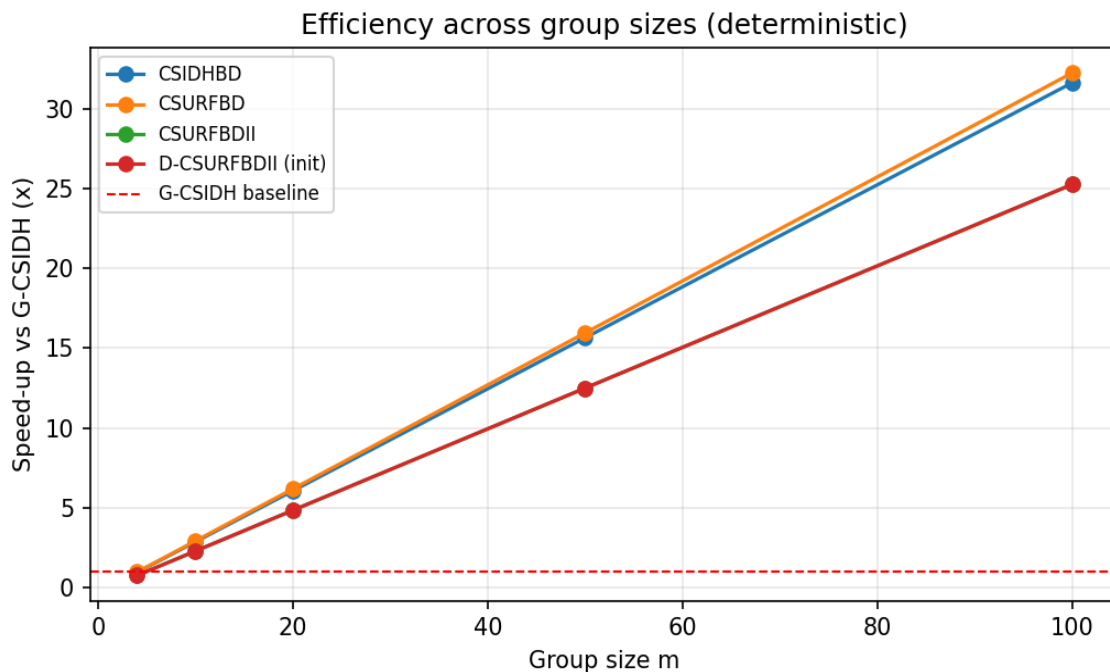


Figure 3. Efficiency across group sizes (deterministic). The multiplicative advantage of the CSURF schemes grows with m ; the static phase of the proposed scheme coincides exactly with CSURFBDII.

The only real uncertainty in this model lies in the calibration constants. We therefore replace the previous jitter-based “sensitivity” study—which varied an injected noise level rather than any property of the protocol—with a sweep over the calibration itself. Table 6 varies the CSURF/CSIDH action ratio across the plausible literature band $[0.95, 1.00]$ and confirms that the qualitative rank order (ring < tree << general) is preserved throughout, and that the CSURF schemes strictly beat their CSIDH counterparts for every ratio below unity. Scaling the light-operation weights across an order of magnitude changes the total cost by less than 0.07%, confirming that the ideal-class action is the decisive cost driver and that the conclusions are insensitive to the light-op calibration.

Table 6. Sensitivity of cost ($m = 20$) to the CSURF/CSIDH action ratio.

CSURF/CSIDH ratio	Tree (CSURFBDII)	Ring (CSURFBD)	General (G-CSURF)	Rank preserved	CSURF < CSIDH
0.95	3.8002	2.9812	18.0500	Yes	Yes
0.96	3.8402	3.0112	18.2400	Yes	Yes
0.97	3.8802	3.0412	18.4300	Yes	Yes
0.98	3.9202	3.0712	18.6200	Yes	Yes
0.99	3.9602	3.1012	18.8100	Yes	Yes
1.00	4.0002	3.1312	19.0000	Yes	n/a

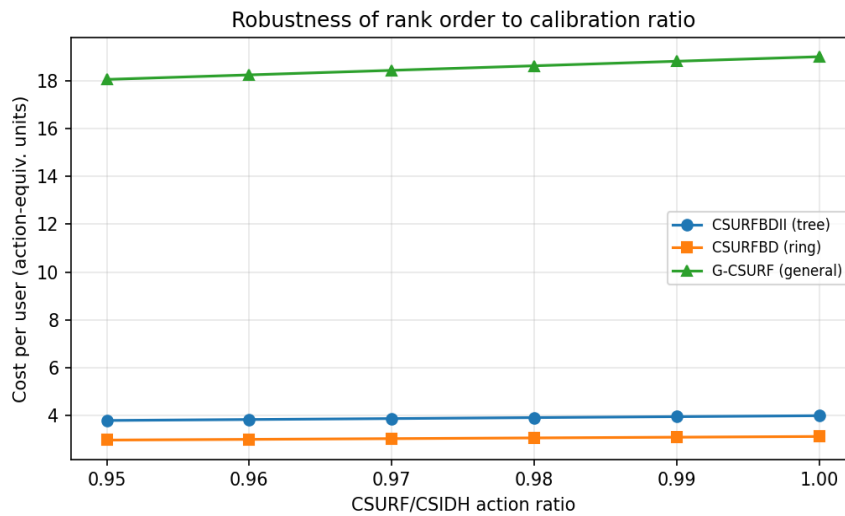


Figure 4. Robustness of the scheme rank order to the CSURF/CSIDH action ratio. The ordering is preserved across the entire plausible band, so the qualitative conclusions do not depend on the precise calibration.

4.7 Distribution of Per-Event Cost over Membership Position

A membership event does not always land at the deepest leaf: its re-key path length equals the depth of the affected node, which varies with where the join or leave occurs in the tree. This positional dependence—not platform jitter—is the genuine source of per-event cost variability in this setting. We therefore report the true distribution of per-event cost over a uniform random event position, summarised by its median and interquartile range together with the best and worst cases, in place of the previous artificially perturbed point estimates.

Table 7. Distribution of per-event cost over a uniform random membership position (action-equivalent units).

m	ℓ min	ℓ max (worst)	Worst-case cost	Min	Q1	Median	Q3	Max
10	1	4	15.6805	3.9205	8.8205	11.7605	14.7005	15.6805
20	1	5	19.6005	3.9205	11.7605	15.6805	16.6605	19.6005
50	1	6	23.5205	3.9205	15.6805	19.6005	23.5205	23.5205
100	1	7	27.4405	3.9205	19.6005	23.5205	27.4405	27.4405
200	1	8	31.3605	3.9205	23.5205	27.4405	31.3605	31.3605

The amortised behaviour over a long-lived session is shown in Figure 6, which plots the running average of per-event cost over a stream of alternating join/leave events landing at random positions for initial groups of $m = 20, 50, 100$. Each curve converges to the position-averaged per-event cost—a stable, group-size-dependent constant—rather than drifting upward, confirming that long sessions do not accumulate overhead. The vertical separation between curves reflects the $O(\log m)$ dependence on group size.

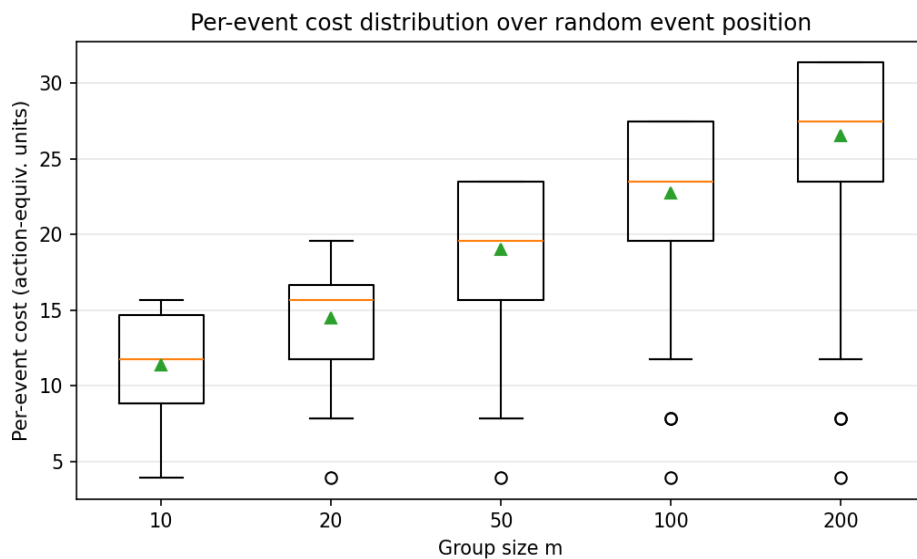


Figure 5. Per-event cost distribution over a random event position, by group size. The spread is the true positional variance of the re-key path length, not injected noise; the median scales logarithmically with m while the worst case remains 40 actions.

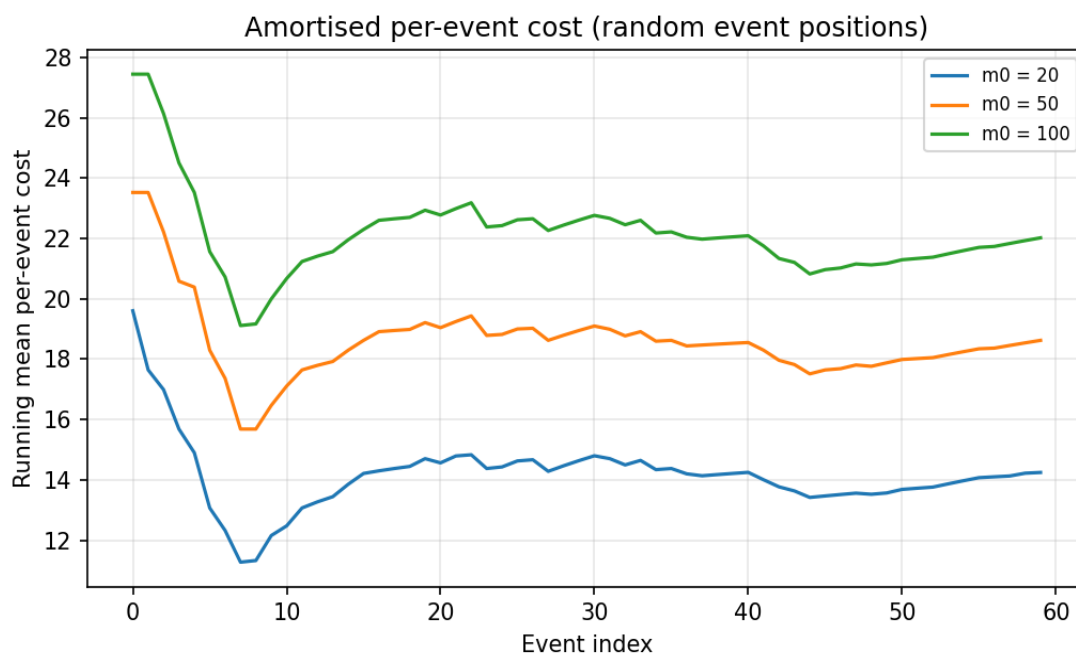


Figure 6. Amortised per-event cost over a random join/leave stream. The running average converges to the position-averaged cost for each group size, demonstrating bounded long-run overhead.

4.8 Threats to Validity

Three caveats bound the interpretation of these results. First, this is an operation-level cost analysis, not a wall-clock benchmark: absolute timings on a constant-time native CSURF implementation may reveal constant factors—particularly memory access and field-arithmetic overheads—that the model abstracts away. The figures should be read as relative complexities, not seconds. Second, the calibration constants are imported from the reference study [16]; Section 4.6 shows the qualitative conclusions survive a sweep over their plausible ranges, but a different calibration would shift absolute magnitudes. Third, the analysis quantifies computation and communication under the passive, unauthenticated, random-oracle model of Section 3; it does not measure the cost of the authentication or post-compromise-healing machinery that an actively secure variant would add. The empirical claims of this paper are therefore claims about operation counts and asymptotic scaling,

deliberately scoped to what the cost model can support. Together, the deterministic tables (3–5), the structural-equality argument (Section 4.4), the closed-form decomposition (Table 4), the calibration sweep (Table 6), and the positional cost distribution (Table 7) support the central thesis without overstating the evidence: D-CSURFBDII inherits the exact static cost profile of CSURFBDII, adds dynamic join and leave at only logarithmic per-event cost, preserves the 2% CSURF advantage, and is the only scheme in the comparison set to provide dynamic membership with forward and backward secrecy.

5. Conclusion

This paper proposed D-CSURFBDII, a dynamic, tree-structured, post-quantum group key exchange protocol built on the CSURF hard homogeneous space. The protocol extends the static CSURFBDII with a dynamic membership manager and a forward/backward-secrecy module that confine re-keying to the logarithmic-length path from an affected leaf to the root of the binary tree. We provided a complete formalization, including epoch-transition equations for join and leave, a correctness proof via the telescoping key identity, and a security reduction to the SCSSDDH assumption in the random-oracle model with advantage bounded by $k(m-1) \cdot \text{Adv}^{\text{CSURF}}$. A comprehensive operation-level evaluation demonstrated that the static phase of D-CSURFBDII is cost-identical to CSURFBDII by construction, that membership events cost only $O(\log m)$, and that the 2% CSURF advantage over CSIDH is preserved. The principal significance of the work is that it is, to our knowledge, the first CSURF-based MPKE to support dynamic membership with provable forward and backward secrecy, closing the gap between efficient isogeny group actions and the logarithmic dynamic re-keying that has hitherto been available only to classical and KEM-based continuous group key agreement. Several limitations qualify these results. First, the evaluation is based on a calibrated operation-level cost model rather than a constant-time, side-channel-hardened native CSURF implementation; the absolute timings are therefore relative rather than wall-clock measurements, and a full implementation may reveal constant factors not captured by the model. Second, the security analysis treats a passive adversary and relies on the random-oracle model; the core protocol is unauthenticated, so an active adversary capable of message injection is outside the present model. Third, the analysis assumes authenticated broadcast channels and an even group size in the basic tree construction, with the odd case handled by a padding convention. Fourth, the forward- and backward-secrecy guarantees are established at the protocol level under honest deletion of stale secrets; post-compromise security in the strong CGKA sense, where an adversary may corrupt and the group must heal, is not formally analyzed. These constitute threats to validity that should be addressed before deployment.

- **Authenticated and actively secure variant:** integrate digital-signature or KEM-based authentication to lift the protocol into the authenticated group key exchange setting and to resist active adversaries, drawing on existing isogeny AGKE frameworks [27].
- **Strong CGKA security:** extend the analysis to post-compromise security and concurrent updates in the continuous group key agreement model, aligning the protocol with the TreeKEM/MLS security standard [21].
- **Constant-time native implementation:** develop a side-channel-resistant CSURF implementation to validate the cost model with wall-clock measurements and to quantify constant factors.
- **Batched and concurrent membership:** design batched join/leave operations that amortize re-keying across multiple simultaneous events without pruning, reducing overhead in high-churn deployments such as the Internet of Vehicles [23].
- **Application integration:** embed D-CSURFBDII into a post-quantum secure messaging or IoT stack and evaluate end-to-end performance against KEM-based CGKA baselines.

References

- [1] M. Steiner, G. Tsudik, and M. Waidner, "Diffie–Hellman key distribution extended to group communication," in *Proc. 3rd ACM Conf. Computer and Communications Security (CCS)*, 1996, pp. 31–37. doi:10.1145/238168.238182.
- [2] M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system," in *Advances in Cryptology – EUROCRYPT 1994*, LNCS, vol. 950, Springer, 1994. doi:10.1007/BFb0053443.
- [3] Y. Kim, A. Perrig, and G. Tsudik, "Simple and fault-tolerant key agreement for dynamic collaborative groups," in *Proc. 7th ACM Conf. Computer and Communications Security (CCS)*, 2000, pp. 235–244.
- [4] D. Vivek, K. Saru, B. Anju, and Y. Sonam, "SL3PAKE: Simple lattice-based three-party password authenticated key exchange for post-quantum world," *J. Inf. Secur. Appl.*, vol. 84, 2024. doi:10.1016/j.jisa.2024.103826.
- [5] S. Furukawa, N. Kunihiro, and K. Takashima, "Multi-party key exchange protocols from supersingular isogenies," in *Proc. Int. Symp. Information Theory and Its Applications (ISITA)*, 2018, pp. 208–212. doi:10.23919/ISITA.2018.8664316.
- [6] L. De Feo, "Mathematics of isogeny based cryptography," *CoRR*, arXiv:1711.04062, 2017.
- [7] D. Jao and L. De Feo, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," in *Proc. Int. Workshop Post-Quantum Cryptography (PQCrypto)*, LNCS, vol. 7071, 2011, pp. 19–34. doi:10.1007/978-3-642-25405-5_2.

-
- [8] R. Azarderakhsh et al., "Supersingular isogeny key encapsulation (SIKE)," Submission to the NIST Post-Quantum Cryptography Standardization Process, 2017. [Online]. Available: <https://sike.org/>.
- [9] W. Castryck and T. Decru, "An efficient key recovery attack on SIDH," in *Advances in Cryptology – EUROCRYPT 2023*, Springer, 2023, pp. 423–447. doi:10.1007/978-3-031-30589-4_15.
- [10] L. De Feo, D. Kohel, A. Leroux, C. Petit, and B. Wesolowski, "SQISign: Compact post-quantum signatures from quaternions and isogenies," in *Advances in Cryptology – ASIACRYPT 2020*, LNCS, Springer, 2020, pp. 64–93. doi:10.1007/978-3-030-64837-4_3.
- [11] P. Dartois, A. Leroux, D. Robert, and B. Wesolowski, "SQISignHD: New dimensions in cryptography," in *Advances in Cryptology – EUROCRYPT 2024*, LNCS, vol. 14651, Springer, 2024. doi:10.1007/978-3-031-58716-0_1.
- [12] W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes, "CSIDH: An efficient post-quantum commutative group action," in *Advances in Cryptology – ASIACRYPT 2018*, LNCS, vol. 11274, Springer, 2018, pp. 395–427. doi:10.1007/978-3-030-03332-3_15.
- [13] W. Castryck and T. Decru, "CSIDH on the surface," in *Post-Quantum Cryptography (PQCrypto 2020)*, LNCS, vol. 12100, Springer, 2020, pp. 111–129. doi:10.1007/978-3-030-44223-1_7.
- [14] X. Fan, X. Xu, and B. Li, "Group key exchange protocols from supersingular isogenies," in *Information Security and Cryptology 2020*, LNCS, vol. 12612, Springer, 2021. doi:10.1007/978-3-030-71852-7_11.
- [15] T. Moriya, K. Takashima, and T. Takagi, "Group key exchange from CSIDH and its application to trusted setup in supersingular isogeny cryptosystems," in *Information Security and Cryptology 2019*, Springer, 2020. doi:10.1007/978-3-030-42921-8_5.
- [16] X. Fan, F. Zhao, and X. Xu, "Multi-party post-quantum key exchange schemes," *J. Inf. Secur. Appl.*, vol. 95, art. 104288, 2025. doi:10.1016/j.jisa.2025.104288.
- [17] W. Castryck, "CSIDH on the surface (CSURF)," Isogeny-Based Cryptography School lecture notes, 2021. [Online]. Available: <https://homes.esat.kuleuven.be/~wcastryck/>.
- [18] National Institute of Standards and Technology, "Post-Quantum Cryptography Standardization: FIPS 203 (ML-KEM), FIPS 204 (ML-DSA), FIPS 205 (SLH-DSA), and selection of HQC," 2024–2025. [Online]. Available: <https://csrc.nist.gov/projects/post-quantum-cryptography>.
- [19] D. X. Charles, K. E. Lauter, and E. Z. Goren, "Cryptographic hash functions from expander graphs," *J. Cryptology*, vol. 22, no. 1, pp. 93–113, 2009. doi:10.1007/s00145-007-9002-x.
- [20] Nature Research Intelligence, "Post-quantum cryptography using elliptic curves and isogenies," Nature Research Intelligence Topic Summary, 2025. [Online]. Available: <https://www.nature.com/research-intelligence/>.
- [21] J. Alwen et al., "Continuous group-key agreement and TreeKEM: concurrent updates without pruning," in *Advances in Cryptology*, Springer, 2023–2025 (MLS / RFC 9420 context). doi:10.1007/978-3-032-01913-4_5.
- [22] R. Schoof, "Nonsingular plane cubic curves over finite fields," *J. Combin. Theory Ser. A*, vol. 46, no. 2, pp. 183–211, 1987. doi:10.1016/0097-3165(87)90003-3.
- [23] G. Xu, X. Yin, and X. Li, "ER-CGKA: Efficient and robust continuous group key agreement scheme with post-compromise forward security for IoT," *PLOS ONE*, vol. 19, no. 8, e0307867, 2024. doi:10.1371/journal.pone.0307867.
- [24] J. Alwen, S. Coretti, Y. Dodis, and Y. Tselekounis, "Continuous group key agreement with active security," in *Theory of Cryptography (TCC)*, LNCS, Springer, 2020. doi:10.1007/978-3-030-64378-2_10.
- [25] S. D. Galbraith, C. Petit, B. Shani, and Y. B. Ti, "On the security of supersingular isogeny cryptosystems," in *Advances in Cryptology – ASIACRYPT 2016*, LNCS, vol. 10031, Springer, 2016, pp. 63–91. doi:10.1007/978-3-662-53887-6_3.
- [26] B. Bencina et al., "An Internet messenger using post-quantum cryptography algorithms based on isogenies of elliptic curves," *Electronics*, vol. 14, no. 14, art. 2905, 2025. doi:10.3390/electronics14142905.
- [27] A. Fujioka, K. Takashima, and K. Yoneyama, "One-round authenticated group key exchange from isogenies," in *Provable Security (ProvSec)*, Springer, 2019, pp. 330–338.
- [28] X. Fan, F. Zhao, L. Wang, J. Fan, and C. Ren, "Post-quantum scheme using Montgomery+ curves," in *Proc. 2024 IEEE 6th Advanced Information Management, Communicates, Electronic and Automation Control Conf. (IMCEC)*, vol. 6, 2024, pp. 307–311. doi:10.1109/IMCEC59810.2024.10575102.
- [29] J. Zhao, D. Gu, and M. C. Gorantla, "Stronger security model of group key agreement," in *Proc. 6th ACM Symp. Information, Computer and Communications Security (ASIACCS)*, 2011, pp. 435–440.
- [30] R. Ishibashi and K. Yoneyama, "Compact password authenticated key exchange from group actions," in *Australasian Conf. Information Security and Privacy (ACISP)*, Springer