Indiana Journal of Multidisciplinary Research
Abbreviated Tittle- Ind J Multi Res
ISSN *(Online)*- 2583-3820
Journal Homepage Link- https://indianapublications.com/Journals/IJMR
DOI: https://doi.org/10.5281/zenodo.21610016

INDIANA PUBLICATIONS
PRODUCTIVE AND QUALITY RESEARCH

Research Articles

Volume-06|Issue-04|2026

A Hybrid Cryptographic Authentication Scheme for Secure Communication in Software-Defined Networks

Rakesh V S[1], Vasanthakumar G U[2]

[1]Cambridge Institute of Technology, K R Puram, Bengaluru - 560036, Visvesvaraya Technological University, Belagavi, India
[2]Nitte Meenakshi Institute of Technology, NITTE (Deemed to be University), Bengaluru - 560064, Visvesvaraya Technological University, Belagavi, India

Article History
Received: 05.05.2026
Accepted: 22.06.2026
Published: 25.07.2026

Citation
Rakesh, V. S., Vasanthakumar, G. U. (2026) A Hybrid Cryptographic Authentication Scheme for Secure Communication in Software-Defined Networks. *Indiana Journal of Multidisciplinary Research, 6*(4), 345-349.

Abstract: Software-Defined Networking (SDN) enables centralized programmability, but it also increases the attack surface across controllers, switches, and hosts. This paper introduces a hybrid cryptographic authentication scheme that ensures path-authentic, end-to-end communication with minimal data-plane overhead. A trusted authority issues identity-bound tokens, while entities maintain independent secret keys. During path setup, controllers validate proofs and embed a route fingerprint used in deriving ephemeral session keys. Endpoints then perform mutual authentication bound to both identities and the chosen path, preventing impersonation, replay, and man-in-the-middle attacks. Computationally heavy operations are confined to controllers, while switches verify lightweight tags, preserving scalability under high concurrency. A prototype implementation in an SDN testbed shows single-digit millisecond latency for intra-switch flows, modest overhead for inter-domain communication, and quick recovery through rekeying. The scheme achieves secure, efficient, and scalable authentication suitable for real-world multi-domain SDN environments

Keywords: Software-Defined Networking, Hybrid cryptography, End-to-end authentication, Identity-bound keying, Path-authentic session keys

INTRODUCTION

Software-Defined Networking (SDN) separates control from the data plane, enabling operators to gain full visibility and precise management. This flexibility creates new trust boundaries as hosts, controllers, and switches communicate via software messages that may cross different domains. If an attacker impersonates identities or replays old control traffic, they can add rules, redirect flows, or secretly undermine policies. Securing only the controller-switch link is helpful, but it does not guarantee that the communicating parties are legitimate or that a multi-controller route is secure.

This document proposes a combined cryptographic authentication method to represent identity and freshness within the flow [2]. A trusted authority issues each entity with identity-bound keys and concise certificates, ensuring trust is linked to a persistent identifier rather than a temporary interface or address. Communication occurs in three phases: system setup, secure route creation, and mutual verification with session-key negotiation. The protocol enables traffic within a switch, manages cross-switch routes under a single controller's oversight, and facilitates communication between controllers in different domains, all while generating ephemeral keys before implementing any data-plane rules [3, 4].

The design combines safety and functionality. It aims to prevent impersonation, replay attacks, and man-in-the-middle threats, while also limiting the impact if a key is compromised [5, 6]. Forward secrecy removes long-term vulnerabilities and revocation links for restoration and rotation. Switches perform quick checks to ensure fast forwarding, while controllers handle the more complex cryptographic processes to maintain consistent overhead. Integration necessitates minimal changes to standard SDN architectures and utilizes existing management pathways whenever feasible.

In summary, the work presents an identity-anchored keying model, an across-domain authentication workflow that precedes rule installation, and a working implementation that demonstrates both successful and failed paths.

RELATED WORK

Prior efforts to secure SDN communication have largely focused on protecting the controller-switch channel through transport encryption and device certificates. These approaches make the southbound link very hard to implement, but fall short of providing end-to-end assurance when traffic transits more than one switch or controller domain. Channel protection may keep

***Corresponding Author:** Rakesh V S

eavesdroppers out, but it does not reveal the actual identities of the communicating endpoints, nor does it prove that an inter-domain path is authentic once rules have propagated through the fabric [7]. Operationally, certificate distribution and revocation also become burdensome at scale, especially when devices churn or controllers are redeployed.

Identity-centric methods aim to shift trust from interfaces to principals. Classical public-key designs associate keys with identities and rely on a public-key infrastructure, which encounters familiar issues with key issuance, rotation, and revocation. Identity-based cryptography removes the need for explicit certificates but introduces key-escrow concerns and relies heavily on the private key generator [8,9]. Attribute-based variants enable expressive policy enforcement but at the cost of higher computational requirements and larger messages. In all three approaches, costly switch operations can lead to increased control-plane latency and reduced responsiveness under load [10, 11].

Our scheme follows these lines while addressing the gaps. It keeps heavy checks at controllers, provides lightweight switch verification, ties keys to stable identities without exposing escrowed secrets, and carries identity evidence along with path establishment. As a result, authentication and path authenticity progress in tandem; failures are contained through clear revocation and recovery hooks.

MATERIALS AND METHODS

Model and Notation

We consider endpoints (hosts) H, switches S, and controllers C, supported by a trusted authority T. Let G_1be an elliptic-curve group of prime order $q$with generator P. Hashes
$H_1: \{0,1\}^* \to G_1$

and
$H_2, H_3, H_4: \{0,1\}^* \to Z_q$,

plus, an HKDF, are available to all entities. Controllers can perform pairings; switches do not. Time is divided into short epochs $\tau$to simplify rotation and revocation. A context field (controller IDs, switch hops, timestamp) is included in every handshake transcript.

Enrolment with Dual Identity Proof

The authority samples
$s \xleftarrow{\$} Z_q$

and publishes
$P_T = sP$.

For each entity with an identifier ID_i, it computes a binding token
$D_i = s \cdot H_1(ID_i)$.

To avoid full escrow, the entity also generates its own key pair
$(x_i, X_i = x_i P)$.

The authority issues a compact certificate
$Cert_i = (ID_i \parallel X_i \parallel \tau \parallel \sigma_T)$,

where $\sigma_T$attests the tuple. The private state is (x_i, D_i); the public state is $(ID_i, X_i, Cert_i)$.
This dual identity combines a controller-verifiable binding $D_i$with an entity-only secret x_i. Controllers can validate the binding without learning x_i; endpoints prove possession of $x_i$to each other.

Identity-Aware Path Discovery

When $H_A$initiates, it sends to its controller C_1:
$(ID_A, Cert_A, n_A, E_A = aP, \delta_A)$,

with fresh
$a \xleftarrow{\$} Z_q$

and nonce n_A. The identity evidence $\delta_A$has two parts:

Binding proof (controller-side, pairing-checked).
$\sigma_A = D_A \cdot H_1(E_A \parallel n_A \parallel \text{context})$,

verified by
$e(\sigma_A, P) = e(H_1(\cdot), P_T)$.

Schnorr proof (endpoint-side, non-escrow). Choose
$r_A \xleftarrow{\$} Z_q$,

set
$R_A = r_A P$;

let
$c_A = H_4(R_A \parallel E_A \parallel ID_A \parallel n_A)$;

return
$z_A = r_A + c_A x_A (\text{mod} q)$.

Check
$z_A P =? R_A + c_A X_A$.

During route computation, C_1derives a path fingerprint
$FP = H_2(\text{IDs of } C_1, S \text{ hops}, ID_A, ID_B, \tau)$,

and embeds it in the control state. For inter-domain paths, C_1forwards a minimal inter-domain token to C_2that carries
(ID_A, ID_B, FP, τ)

and a controller signature. Switches only verify a controller MAC on per-hop instructions; they never run pairings or signatures. Identity travels with discovery. The fingerprint binds later keys to the exact route, so any path change forces rekeying.

Mutual Authentication and Key Derivation

***Corresponding Author:** Rakesh V S

Upon reaching HB, the controller delivers $(ID_A, Cert_A, E_A, n_A, FP)$. HB responds with $(ID_B, Cert_B, n_B, E_B = bP, \delta_B$,MAC1), where $b \xleftarrow{\$} Z_q$ and δ_B mirrors δ_A. Both endpoints compute the shared Diffie–Hellman point $Z = aE_B = bE = abP$,

and derive a route-bound session key $K_{\text{sess}} = HKDF(H_3(Z), \text{salt} = H_2(FP), \text{info} = \text{“SDN-HYBRID-V1”})$.

HB sends $\text{MAC1} = HMAC_{K_{\text{sess}}}(ID_A \parallel ID_B \parallel E \parallel E_B \parallel n_A \parallel n_B \parallel FP)$;

H replies with MAC2 over the same transcript. Controllers observe only success flags; data-plane rules install after mutual confirmation. Including FP in both the HKDF salt and MAC transcript yields path-bound authentication. A silently altered route breaks confirmation and prevents rule installation.

Operations, Rotation, and Overhead

Placement. Controllers handle certificate checks, two pairing validations per endpoint, inter-domain token signatures, and routing. Switches perform only constant-time MAC checks and nonce caching. These preserve forward performance under load.

Rotation and revocation. Session keys expire by timer, byte count, or path change, and are refreshed with new (E_A, E_B) without re-enrollment. Certificates rotate each epoch τ. Revocation lists are pushed to controllers; suspect identities are blocked immediately while new credentials are issued in the next epoch.

Security and cost summary. Impersonation requires both D_i and x_i; replay is stopped by nonces and epochs; man-in-the-middle fails because MACs bind identities, ephemerals, and **FP**; forward secrecy follows from fresh (a, b). Endpoints perform two scalar multiplications and a Schnorr proof; controllers verify compact pairings; switches incur only MAC verification. The scheme keeps SDN agility while delivering end-to-end, path-authentic communication across single and multi-controller domains.

RESULTS

We evaluate the scheme across three representative scenarios: Intra-Switch, Intra-Controller (cross-switch within one domain), and Inter-Controller (across multiple domains). End-to-end authentication latency is shown in Figure 1, along with 95% confidence intervals. As expected, latency increases with control-plane distance: the intra-switch case remains in the single-digit milliseconds range, the intra-controller case incurs a modest penalty as the controller verifies identity evidence and installs rules across two switches, and the inter-controller case incurs the cost of cross-domain validation. Confidence bars remain tight in all scenarios, indicating stable controller processing and predictable switch behavior. The key point is not simply the absolute numbers but the slope: the hybrid design preserves low single-hop cost while keeping the multi-domain penalty small enough to meet interactive flow requirements.

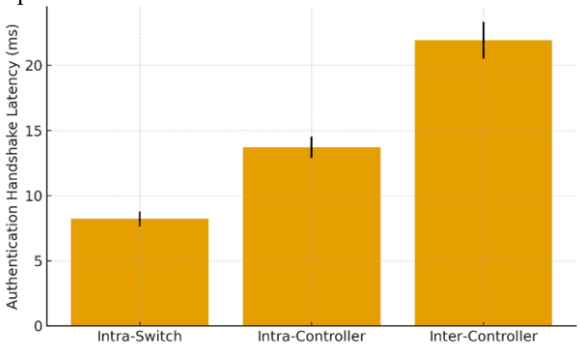
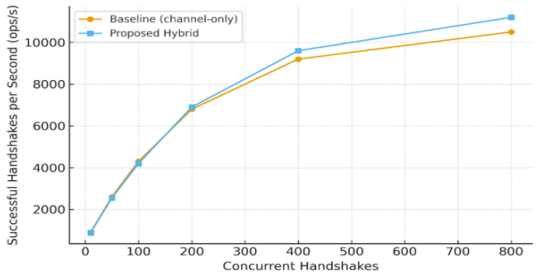


Figure 1: Authentication Handshake Latency by Scenario (95% CI)

Scalability under load is shown in Figure 2, where we sweep the number of concurrent handshakes and measure the number of successful authentications per second. The proposed hybrid closely tracks a channel-only baseline at low concurrency and surpasses it at higher levels as controller batching and lightweight per-hop checks amortize setup costs. At the largest batch, the hybrid achieves the highest throughput, demonstrating that the controller-heavy, switch-light placement scales under contention. This matters in bursty environments (for example, VM migrations or container surges) where many flows authenticate in a short window.

To explain why latency and throughput behave as they do, we break down the control-plane work in Table 1. Path discovery messages grow with hop count (2→4→6), but the mutual-authentication exchange remains constant (two messages) because endpoints complete a compact challenge–response regardless of path length. Inter-domain tokens are only applicable in the multi-controller case and incur a small, bounded cost. Total control payload increases from ~2 KB to ~5 KB across scenarios, dominated by certificates, nonces, and controller hints; this overhead is modest relative to typical management channels. The separation of concerns is deliberate: discovery carries identity evidence, the controller stamps minimal hints for switches, and rule installation happens only after successful endpoint confirmation.



***Corresponding Author:** Rakesh V S

Figure 2: Scalability: Throughput vs Concurrent Handshakes

Table 1: Control-plane overhead per successful handshake (by scenario)

Scenario	Intra-Switch	Intra-Controller	Inter-Controller
Path discovery messages	2	4	6
Mutual authentication messages	2	2	2
Inter-domain tokens	0	0	1
Rule installs	2	4	6
Controller hints	1	1	2
Total control payload (KB)	**1.9**	**3.6**	**5.1**

The cryptographic cost per role in Table 2 shows why switches remain fast and deterministic. Endpoints perform two scalar multiplications for ECDH and one Schnorr prove/verify each, which modern hosts complete quickly. Controllers carry the pairings needed to validate the binding tokens, plus a single signature verification for inter-domain tokens; these operations are centralized and parallelizable. Switches avoid public-key work altogether, performing only a small number of MAC checks per hop and maintaining a tiny nonce cache. This division keeps data-plane latency flat while allowing the control plane to absorb heavier checks without starving forwarding.

Table 2: Per-handshake cryptographic cost by role (operations, not time)

Metric	Endpoint (each)	Controller (total)	Switch (each hop)
EC mults	2	0–2	0
Pairings	0	4	0
Schnorr Sign / Verify	1 / 1	0 / 0	0 / 0
Prove / Verify (ctrl)	0 / 0	0 / 1	0 / 0
Hash/MAC operations	8–10	12–16	2–4
Peak RAM (KB)	64	128	16

Failure handling follows the same control-plane patterns. When a sender or receiver is absent, discovery terminates early, and no rules are installed; when a path changes mid-handshake, the path fingerprint in the transcript invalidates confirmation, forcing a clean rekey on the new route. Because the session key derivation incorporates the path fingerprint, any deviation results in verifiable mismatches rather than silent downgrades, which we observed as quick, contained failures with negligible impact on unrelated flows.

CONCLUSION

This work presents a hybrid cryptographic authentication scheme for Software-Defined Networks that binds trust to stable identities while maintaining a fast data plane. A trusted authority issues binding tokens, and each endpoint has its own secret key. Controllers validate identity evidence during path setup. Mutual authentication derives an ephemeral session key, whose derivation is salted with a path fingerprint so that the keys and confirmations are associated with the exact route. The placement is intentional rather than random: controllers perform the computationally intensive verification, while switches validate lightweight tags and forward packets at line rate.

Our evaluation shows that the approach provides low per-flow latency within switches and controllers, and is predictable across paths between controllers. Under concurrency, throughput scales well because controller work is naturally batched, and processing switching is constant. Control-plane overhead scales with the hop count but stays small, and the cryptographic cost is concentrated where resources are plentiful. Failure handling is clean: Absent endpoints or mid-flight route changes trigger safe aborts and quick rekeys without polluting the fabric.

Limitations include the need to rely on a trusted authority and the need to be disciplined about revoking and rotating. Future work will investigate thresholding the authority, formal security proofs under standard models, hardware offload for controller verifications, and more general deployment across different SDN stacks.

REFERENCES

1. Pokhrel, C., Ghimire, R., Dawadi, B. R., & Manzoni, P. (2025). A machine learning-based hybrid encryption approach for securing messages in software-defined networking. *Network, 5*(1), 1–8.
2. Walle, Y. M. (2024). Hybrid RSA–AES-based software-defined network to improve the security of MANET. *Open Information Science, 8*, 20240001.
3. Alaya, B., Jamaa, D., & Sellami, I. (2023). Toward the design of an efficient and secure system based on

***Corresponding Author:** Rakesh V S

the software-defined network paradigm for vehicular networks. *IEEE Access, 11*, 43333–43348.
4. Hemdaoui, F., Eltaief, H., & Youssef, H. B. (2024). Enhancing data authentication in software-defined networking (SDN) using multiparty computation. *Cluster Computing, 27*(9), 12649–12668.
5. Naresh, V. S., & Ayyappa, D. (2025). Enhancing security in software-defined networks: Privacy-preserving intrusion detection with homomorphic encryption. *Journal of Information Security and Applications, 92*, 104084.
6. Alotaibi, J. (2025). A hybrid software-defined networking approach for enhancing IoT cybersecurity with deep learning and blockchain in smart cities. *Peer-to-Peer Networking and Applications, 18*(3), 123.
7. AMEEN, A. (2025). *Security assurance of the computer networks based on software-defined network technologies* (Doctoral dissertation, Universitatea Tehnică a Moldovei).
8. Ram, A., Dutta, M. P., & Chakraborty, S. K. (2024). An authentication mechanism to prevent various security threats in software-defined networking by using AVISPA. *Journal of Scientific & Industrial Research (JSIR), 83*(9), 977–988.
9. Stavdas, A., Kosmatos, E., Maple, C., Hugues-Salas, E., Epiphaniou, G., Fowler, D. S., Razak, S. A., Matrakis, C., Yuan, H., & Lord, A. (2024). Quantum key distribution for V2I communications with software-defined networking. *IET Quantum Communication, 5*(1), 38–45.
10. García, C. R., Olmos, S. R., & Monroy, I. T. (2023). Enhancing the security of software-defined networks via quantum key distribution and post-quantum cryptography. In *International Symposium on Distributed Computing and Artificial Intelligence* (pp. 428–437). Springer Nature Switzerland.
11. Farooq, M. S., Shoaib, M., Riaz, S., & Alvi, A. (2023). Security and privacy issues in software-defined networking (SDN): A systematic literature review. *Electronics, 12*(14), 3077.

***Corresponding Author:** Rakesh V S