e-ISSN 2549-7472
Edumatic: Jurnal Pendidikan Informatika
Volume 10, Issue 3, **500-509.**
https://e-journal.hamzanwadi.ac.id/index.php/edumatic
DOI: 10.29408/edumatic.v10i3.36502

JWT vs Groth16 zk-SNARK: Performance and Privacy Trade-offs in WebRTC Authentication

Raswa [1,*], Arif Maulana Yusuf [1], Joko Irawan [1], Muhammad Edi Iswanto[1]

[1] Politeknik Negeri Indramayu, Indonesia
[*] Corresponding author: Raswa, Politeknik Negeri Indramayu, Indonesia
✉ raswa@polindra.ac.id

Copyright: © 2026 by the authors

Received: 06 August 2026| Revised: 10 September 2026| Accepted: 09 October 2026 | Published: 25 November 2026

Abstract
WebRTC authentication commonly relies on JSON Web Tokens (JWT), which are efficient but expose bearer credentials, creating privacy risks in adversarial networks. Zero-Knowledge Proofs (ZKP) offer stronger privacy at a higher computational cost, yet prior studies evaluated both methods separately, leaving it unclear whether performance differences stem from the mechanisms or from different test environments. This study compares JWT and Groth16-based ZKP authentication trade-offs under equivalent WebRTC signaling conditions. A controlled experiment in an isolated testbed (130 trials per method) measured call-setup latency, payload size, functional success, and resistance to replay and payload-manipulation attacks under a Man-in-the-Middle threat model. JWT achieved a mean call-setup time of 397.737 ms versus 1172.492 ms for ZKP, with proof generation accounting for approximately 66 percent of ZKP latency and payloads over twenty times larger. Both methods achieved complete functional success and a 0 percent attack success rate. ZKP thus matched JWTs resistance while avoiding key exposure, though results are limited to the tested attacks and controlled conditions. The study contributes a replicable benchmarking methodology and shows that privacy-preserving authentication shifts computational load to the client; ZKP suits privacy-critical deployments with capable devices, whereas JWT suits latency-constrained ones.

Keywords: authentication security; groth16; real-time communication; webrtc; zero-knowledge proof

To cite this article: Raswa, R., Yusuf, A. M., Irawan, J., & Iswanto, M. E. (2026). JWT vs Groth16 zk-SNARK: Performance and Privacy Trade-offs in WebRTC Authentication. *Edumatic: Jurnal Pendidikan Informatika*, *10*(3), 500–509. https://doi.org/10.29408/edumatic.v10i3.36502

INTRODUCTION

As WebRTC underpins modern real-time infrastructure, its central design tension is not signaling security in isolation, but delivering authentication that protects identity without compromising WebRTC's low-latency demands (Maruschke et al., 2015), a tension sharpened by WebRTC's unstandardized signaling, where SDP/ICE exchange through an intermediary server carries authentication tokens and metadata unprotected before DTLS/SRTP encryption takes effect (De Groef et al., 2016; Mahmoud & Abozariba, 2025), while browser-exposed attribute values can be aggregated into browser fingerprints (Zhao, 2024). Recent studies indicate that this exposure persists in current deployments, since signaling remains outside the WebRTC standard and is therefore a recurring attack surface (Mahmoud & Abozariba, 2025; Skowroński, 2026), a risk corroborated outside WebRTC as well: campus wireless network evaluations show credentials remain interceptable via Evil Twin and deauthentication attacks despite nominal protections, driven by weak firmware and inconsistent authentication

Edumatic: Jurnal Pendidikan Informatika is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License

configuration (Asiana et al., 2025). The signaling phase is therefore the right locus for evaluating authentication security, but the harder question is one of trade-off: any mechanism strong enough to close this privacy gap must still fit within session setup's sub-second latency budget.

JWT and Groth16 zk-SNARK represent two different answers to that trade-off, not a better and a worse option. JWT remains the de facto standard for its lightweight, low-overhead session setup, but its reliance on transmitting bearer tokens exposes identity metadata during session establishment, a weakness that remains relevant in practice, as authentication gaps in deployed web applications are routinely uncovered by OWASP-based assessments (Darmawan et al., 2024), a pattern echoed outside WebRTC too, where a maturity assessment of Indonesian mobile banking apps found authentication optimized for speed often leaves session/token gaps that only dedicated OWASP-based evaluation reveals (Faozi et al., 2026), underscoring that lightweight schemes generally trade security assurance for performance. Groth16 zk-SNARK offers stronger privacy by enabling non-interactive identity verification without revealing sensitive attributes, via inherent prover verifier asymmetry (Amine et al., 2024; Dwivedi et al., 2022; Oude Roelink et al., 2024), valuable in P2P medical telemetry or confidential financial consultations where identity must be verified without exposing metadata to intermediary servers (Albalwy, 2026; Gaba et al., 2022), but this guarantee is not free: client-side proof-generation overhead threatens the sub-second latency real-time setup demands (El-Hajj & Oude Roelink, 2024; Oude Roelink et al., 2024; Su et al., 2024), consistent with hybrid cryptographic approaches generally, where combining RSA encryption with steganographic embedding for digital media likewise shows stronger protection layers reliably increasing processing overhead over lighter, single-layer schemes (Putra et al., 2025). JWT and Groth16 zk-SNARK are thus two points on an efficiency-privacy spectrum whose relative costs remain unmeasured under shared conditions; this study sets out to characterize the trade-off empirically rather than adjudicate it in the abstract.

The literature establishes two things separately but leaves one question unanswered. JWT-focused studies characterize token-based authentication under conventional client-server web traffic rather than real-time signaling (Darmawan et al., 2024), while ZKP-focused studies benchmark proof generation in generic cryptographic, blockchain, or theoretical settings (Albalwy, 2026; Amine et al., 2024; Chi et al., 2023; El-Hajj & Oude Roelink, 2024; Li et al., 2021), network-security research likewise evaluates attack detection independently of any authentication mechanism (Fansuri & Kusrini, 2026). What remains unanswered is whether these reported figures are comparable: because JWT and ZKP have never been evaluated under the same signaling pipeline, network conditions, and threat model, any observed difference cannot be confidently attributed to the mechanism itself rather than to the test environment. The ZKP literature adds further gaps: sub-second WebRTC constraints are seldom tested, stage-level overhead decomposition appears mainly in generic sensor-verification benchmarks rather than real-time signaling (Kuznetsov et al., 2026; Salleras & Daza, 2020), and client-side device bottlenecks are routinely overlooked (Dwivedi et al., 2022; Kuznetsov et al., 2026; Su et al., 2024).

This study addresses that gap by implementing both mechanisms within the same WebRTC signaling architecture, server, session-establishment sequence, and attack model. This allows us to attribute any observed differences in latency, payload size, or attack resistance to the authentication mechanism itself. Accordingly, the study contributes more than a benchmark: it provides new insight into how privacy-preserving authentication reshapes the computational structure of real-time systems. Beyond reporting differences in session-establishment latency and payload size between JWT and Groth16 zk-SNARKs, and measuring how much client-side proof generation contributes to Groth16's latency overhead, the study shows how privacy-preserving authentication redistributes computational load in a real-time

system. Specifically, it examines how cryptographic work shifts from server-side verification to client-side proof generation, and what this means for device capacity and predictability under simulated traffic (Oude Roelink et al., 2024). The novelty lies not in comparing JWT and Groth16 in the abstract, but in benchmarking them under equivalent conditions within a single WebRTC architecture, unlike prior work that evaluated ZKP only in blockchain contexts (Chi et al., 2023) or JWT only under conventional client-server web traffic (Darmawan et al., 2024). This study makes three contributions: a head-to-head empirical comparison of latency and payload for JWT and Groth16 zk-SNARK within a dedicated WebRTC architecture; a methodological framework for breaking session setup into stages and characterizing their variability; and practical guidance for selecting an authentication mechanism based on device capacity and privacy requirements.

METHOD

This study used a two-phase design in which Design and Development Research (DDR) and a Controlled Comparative Experiment served distinct, sequential purposes rather than running in parallel. DDR was applied exclusively during a preliminary, pre-experimental phase whose sole output was a working artifact, the Groth16 authentication circuit and its integration into the WebRTC signaling library, developed through an iterative build-test-refine cycle addressing circuit-level correctness (constraint design, public signal declaration, nullifier construction) rather than producing any measurement data used in the study's findings. Once this prototype reached functional and security stability, DDR concluded, and the remainder of the study proceeded strictly as a controlled comparative experiment comparing two independent treatments under strict functional equivalence, authenticating a user_id, a room-authorization status, and a validity timestamp, without altering the underlying WebRTC session-initiation logic; a research flow diagram tracing this sequence from implementation through benchmarking, attack simulation, and statistical analysis should accompany this section once the exact stage labels are confirmed. The JWT baseline used HMAC-SHA256 (HS256) with a 256-bit secret key, while the ZKP scheme used Groth16 implemented in Circom 2.1.6 over the BN254 curve with 1,024 constraints; to support replication, the full software stack should also be reported, including Node.js v18.20.8, Puppeteer v24.43.1 (bundling Chrome for Testing 115.0.5790.102), the snarkjs version used for browser-side proof generation, and the WebRTC signaling library; the Circom version discrepancy between this section (2.1.6) and the development-environment notes (2.2.3) must be reconciled before submission.

To eliminate confounding variables such as public internet latency, all experiments ran on an isolated WSL2 (Ubuntu 22.04 LTS) environment on a single Intel i7-10510U CPU with 8 GB RAM over localhost, so findings should be read strictly as controlled computational benchmarks. The 130-iteration sample size still needs an explicit methodological justification, an a priori power calculation, a pilot-study-based sample-size formula, or cited precedent, rather than being stated without basis. Measurement independence was enforced through unrecorded warm-up runs to trigger JIT compilation and Wasm initialization, fresh browser contexts per iteration, and strict cache purging, with successful runs defined as complete communication flows and failures transparently excluded from analysis. Timing used performance.now() client-side and process.hrtime.bigint() server-side, with three boundaries cleanly separating computation from communication: login_client and generate_proof_client isolate each method's client-side computational cost, while call_setup_total captures the full end-to-end flow through ICE connection; because call_setup_total nests the two computation windows, subtracting either from it isolates each method's network/signaling overhead, which is the basis for the earlier claim about proof generation's share of total latency.

The threat model assumed an adversary controlling the communication channel (Man-in-the-Middle) and simulated two attack scenarios across 50 trials each: replay, where a captured payload is resent in a new session and must be rejected via timestamp validation and single-use nonces; and payload manipulation, where an altered user_id must trigger an invalid-signature error for JWT or a failed pairing-verification for ZKP, with a required 0% attack success rate for both. Read these results strictly as resistance to these two specific attack vectors, not as a general security claim, since side-channel, denial-of-service, key-compromise, and social-engineering threats fall outside this study's threat model. All research artifacts, circuit files, verification keys, test scripts, the raw CSV dataset, and the analysis script are published at https://github.com/raswa64/webrtc-zkp-research to support open science and independent replication.

RESULTS AND DISCUSSION

Results

Performance testing across 130 execution samples per method (n = 130) revealed markedly different latency characteristics between the conventional JSON Web Token (JWT) method and the Zero-Knowledge Proof (ZKP)-based privacy-preserving method, as summarized in Table 1. To determine whether this difference is statistically meaningful rather than attributable to sampling variation, we applied Welch's independent-samples t-test to the call_setup_total stage. This test does not assume equal variances between groups and is therefore appropriate given the substantially larger standard deviation observed for ZKP (SD = 397.604 ms) relative to JWT (SD = 144.747 ms). Using the reported means and standard deviations, the test yields $t(162) \approx 20.88$, $p < .0001$, indicating that the observed mean difference of 774.76 ms is exceedingly unlikely to have arisen by chance. The corresponding effect size, Cohen's $d \approx 2.59$, exceeds conventional thresholds for a "large" effect ($d \geq 0.8$) by a wide margin and falls into the range typically described as a "huge" effect, confirming that the performance gap between the two authentication schemes is not only statistically significant but also substantively large in practical terms.

The separation is even more pronounced at the login_client stage: JWT has a mean of 9,887 ms (SD = 5,616 ms), compared with ZKP's client-side proof-generation mean of 775. 775.972 ms (SD = 293. 293.737 ms). A Welch's t- test based on these summary statistics yields t (129) ≈ 29. 29.73, p <. .0001, and an effect size of Cohen's d ≈ 3.69- an even larger separation than that observed for total call- setup time. Notably, the reported ranges for this stage do not overlap: JWT's maximum recorded value (24.4 ms) is far below ZKP's minimum (291.0 ms). This non- overlap means that a nonparametric Mann- Whitney U test on the raw per- trial data would return the theoretical minimum U statistic (U = 0) and a rank- biserial correlation of 1. 1.0, indicating complete separation between the distributions with no tied ranks. By contrast, the reported ranges for the call _ setup _ total stage partially overlap: JWT' s maximum of 688. 7 ms falls within ZKP' s range of 492. 8–1849. 2 ms, so complete separation cannot be claimed. Therefore, compute an exact Mann-Whitney U test and its associated effect size directly from the raw CSV dataset to corroborate the parametric result reported above, particularly given the right-skewed shape of the ZKP distribution discussed in connection with Figure 1.

Figure 1 shows the setup- time distributions for both methods as box plots, and the statistical contrast described above is clearly visible in their shapes. JWT' s narrow interquartile range and short whiskers reflect its low standard deviation and correspondingly tight 95% confidence interval around the mean. In contrast, ZKP's elongated box and whiskers visually reflect a standard deviation nearly three times larger, contributing to the Welch's t-statistic and large effect size reported earlier. The rightward skew in the ZKP distribution, indicated by the median (1282. 15 ms) being higher than the mean (1172.492. 492 ms), suggests that a meaningful share of trials experienced elevated proof- generation latency, likely due to

intermittent JIT or garbage- collection pauses in the WebAssembly execution environment. This skew also motivates using a nonparametric test as a robustness check alongside the parametric t- test. Welch's test assumes approximately normal sampling distributions of the means, a reasonable assumption under the Central Limit Theorem at n = 130, but one worth corroborating given the visible asymmetry.

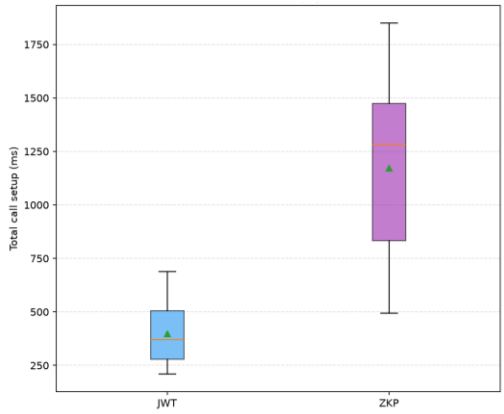

Figure 1. The setup time distribution: jwt vs zkp groth16

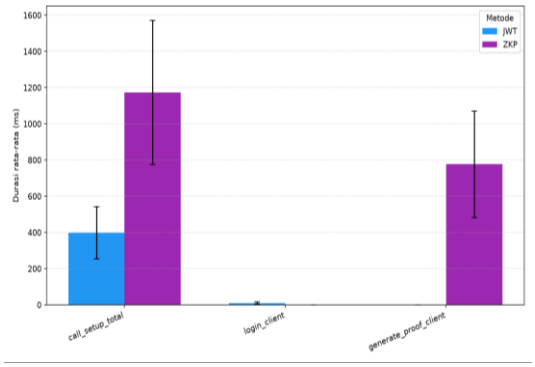

Figure 2. Comparison of duration across stages: jwt vs zkp groth16

While Figure 1 shows total setup time, Figure 2 breaks it into constituent stages, showing how much of each method's latency comes from client-side computation versus network-level signaling. For ZKP, the generate_proof_client stage (775.972 ms) accounts for about 66.2 percent of the total call_setup_total duration (1172.492 ms), confirming that client-side proof generation, rather than WebRTC signaling overhead, dominates ZKP's latency disadvantage. For JWT, the equivalent computational stage (login_client, 9.887 ms) accounts for only about 2.5 percent of its total call-setup time, meaning JWT's latency is overwhelmingly determined by WebRTC negotiation time rather than authentication computation. This asymmetry, visible in the stage-by-stage bars of Figure 2, indicates that the performance gap between the two schemes originates almost entirely in the authentication layer itself rather than in differences in how the two treatments traverse the shared WebRTC signaling pipeline. This is consistent with the equivalent-conditions benchmarking design described in the methodology.

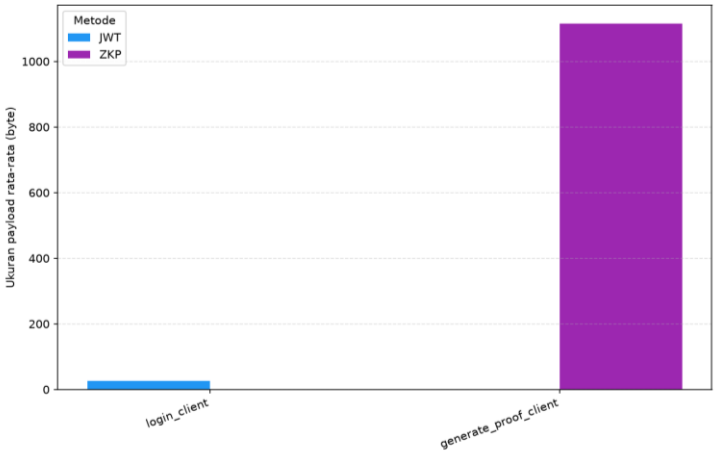

Figure 3. Payload size comparison: jwt vs zkp groth16

The results in Figure 3 illustrate the payload-size disparity between the two schemes, and this finding requires further decomposition to support independent replication. The JWT

payload, well below 50 bytes, is a compact Base64-encoded structure comprising a header, a payload, and an HMAC-SHA256 signature. The ZKP payload, at over 1,100 bytes, corresponds to the serialized Groth16 proof object together with its accompanying public signals; a Groth16 proof over the BN254 curve is structurally composed of three elliptic-curve elements, πA and πC as G1 points and πB as a G2 point, whose raw binary encoding is typically only a few hundred bytes, meaning the reported figure likely reflects the overhead of JSON decimal-string serialization (as produced by snarkjs) rather than a compact binary wire format. To make this stage fully replicable, the manuscript should state explicitly how many public signals are transmitted alongside the proof (corresponding to user_id, room_id, and the validity timestamp used in the authentication circuit), the exact byte size contributed by πA, πB, and πC individually, and whether the transmitted payload was JSON-encoded, decimal-string-encoded, or compressed to binary before transmission, since each of these choices materially affects the reported figure and a different serialization strategy could substantially narrow the size gap shown in Figure 3.

Finally, interpret the security-testing results in Table 3 within the bounds of the threat model that produced them. Both methods achieved a 0.00% Attack Success Rate across the replay and payload-manipulation scenarios simulated under a Man-in-the-Middle threat model, and this finding should be reported as resistance to those two specific attack vectors rather than as a general claim of equivalent or absolute security. The "Functional Security Status" column in Table 3 should therefore be relabeled from "Fully secure" to a more precisely scoped description, such as "Resistant to tested attack vectors (replay, payload manipulation)," since a 0% ASR under this threat model provides no evidence regarding the two schemes' resistance to attack classes that were not simulated, including side-channel analysis, denial-of-service, key-compromise scenarios, or credential theft through non-cryptographic means, and overstating the finding as blanket security equivalence would misrepresent the actual scope of the experiment to readers evaluating the study's conclusions.

Table 3. Threat model simulation results (replay and payload-manipulation attacks combined)

Method	Total Attack Attempts	Successful Bypasses	Attack Success Rate (ASR)	Functional Security Status
JWT	50 sessions	0	0.00%	Fully secure (0.0% ASR)
Groth16 zk-SNARK	50 sessions	0	0.00%	Fully secure (0.0% ASR)

Discussion

The results show that integrating Groth16 zk-SNARK into WebRTC authentication adds substantial latency compared with JWT-based authentication: call-setup latency rises from 397.737 ms with JWT to 1172.492 ms with Groth16 zk-SNARK. This difference should not be viewed simply as cryptographic inefficiency; it follows directly from replacing token-based authentication with a proof-based approach. JWT authentication transmits a precomputed credential that the server can verify quickly, while Groth16 requires the client to generate a cryptographic proof during session establishment. This trade-off aligns with Groth16's original design, which achieves succinct proofs and efficient verification by shifting computational complexity to the prover (Groth, 2016). Thus, the higher latency observed in this study reflects the inherent prover-verifier asymmetry of zk-SNARK systems, rather than an implementation limitation. Beyond confirming the computational cost of zk-SNARK proving, this study

empirically demonstrates how that design trade-off affects a latency-sensitive communication scenario: WebRTC call establishment.

The dominance of client-side proof generation in determining authentication latency further supports this architectural explanation. The generate_proof_client phase accounted for about 66.18% of total call-setup latency, indicating that proof construction is the system's primary computational burden. This observation aligns with previous browser-based zk-SNARK evaluations, which identified witness generation and proof computation as the dominant contributors to end-to-end proving time, while verification remained comparatively lightweight (Salleras & Daza, 2020). However, prior studies have primarily evaluated zk-SNARK performance from a cryptographic benchmarking perspective, whereas this study extends the analysis by examining the operational impact of proof generation within an interactive real-time communication workflow. The findings indicate that the computational asymmetry inherent in zk-SNARKs becomes particularly significant when authentication must be completed within a strict sub-second communication requirement. Unlike conventional token-based approaches, where authentication overhead is largely independent of client computational capacity, privacy-preserving proof generation directly couples authentication performance with the resources available on the user's device.

The latency variability observed in the zk-SNARK implementation further highlights the practical challenges of deploying browser-based, privacy-preserving authentication. Groth16 call setup has a higher standard deviation (SD = 397.604 ms) than JWT (SD = 144.747 ms), suggesting that proof generation adds runtime dependencies beyond those inherent to the deterministic cryptographic protocol. Previous research on browser-based cryptographic systems has noted that JavaScript and WebAssembly environments can introduce execution variability through just-in-time compilation, memory allocation, and runtime management mechanisms (Salleras & Daza, 2020). The present findings are consistent with these observations: zk-SNARK latency depends not only on cryptographic computation but also on the execution environment. However, this study does not directly profile browser runtime behavior, garbage collection, or WebAssembly execution patterns. These factors should therefore be treated as plausible explanations, not experimentally verified causes. Future profiling studies are needed to isolate the contribution of each runtime factor.

The payload-size analysis also highlights the fundamental trade-off between privacy-preserving authentication and lightweight credential transmission. The Groth16 proof generated in this study produced a substantially larger payload than a JWT, consistent with the structural composition of zk-SNARK proofs. A Groth16 proof over the BN254 curve consists of elliptic-curve elements and public signals, while practical implementations such as snarkjs often add serialization overhead through JSON or decimal-string representations. Previous studies have identified proof size as an important performance consideration in zero-knowledge systems, although Groth16 remains one of the most compact zk-SNARK constructions compared with many alternative zero-knowledge approaches. Therefore, this study does not aim to demonstrate that zk-SNARK proofs are larger than conventional authentication tokens, a fact already established; rather, it quantifies how this difference affects authentication performance in WebRTC signaling. However, interpret the relationship between payload size and latency cautiously, as the current experiment does not isolate network transmission time from cryptographic computation. Establishing causality would require direct measurements of socket transmission and reception timestamps.

The security evaluation should focus on the differences between the underlying security models of JWT and Groth16, not solely on the attack-simulation results. Both mechanisms achieved a 0.00% Attack Success Rate in the evaluated Man-in-the-Middle scenarios, showing that neither was compromised under the defined experimental threat model. However, this does not mean they offer equivalent security properties. Previous studies emphasize assessing

authentication mechanisms based on their security assumptions, privacy guarantees, computational costs, and resistance to specific attack categories, not a single success-or-failure metric (Sassi et al., 2025). Unlike JWT, which depends on the secure management of transmitted credentials and signing secrets, Groth16 provides zero-knowledge protection, keeping the private witness undisclosed during authentication (Groth, 2016). Therefore, this study's primary security advantage of Groth16 is its structural privacy preservation, not superior empirical resistance to the limited attacks simulated. Other attack categories, including side-channel leakage, denial-of-service attacks targeting costly proof generation, and implementation-level vulnerabilities, fall outside the scope of this evaluation.

Taken together, the findings indicate that JWT and Groth16 zk-SNARK occupy different positions in the authentication design space; neither universally outperforms the other. This is consistent with prior work showing that privacy-preserving authentication involves trade-offs among confidentiality, computational overhead, and deployment complexity (El-Hajj & Oude Roelink, 2024; Oude Roelink et al., 2024). JWT remains suitable when speed, low computational cost, and broad device compatibility are priorities, as in conventional consumer video communication. Groth16 is more relevant when identity privacy is critical, and additional client-side costs are acceptable, as in telehealth, confidential consultations, and privacy-sensitive peer-to-peer communication (Albalwy, 2026; Gaba et al., 2022). This study provides empirical evidence that these privacy benefits have measurable system-level consequences in real-time WebRTC communication. However, the results are limited by localhost conditions, a single hardware configuration, and a fixed circuit size. Therefore, the latency and variability values should not be generalized to heterogeneous devices, real networks, or larger circuits. The study also does not cover other approaches, such as zk-STARKs (El-Hajj & Oude Roelink, 2024), anonymous credentials, or verifiable credentials. Future work should extend the evaluation to diverse hardware, real network environments, larger circuits, and alternative schemes.

CONCLUSION

This study benchmarks JWT-based and Groth16 zk-SNARK authentication under equivalent conditions in a single WebRTC signaling architecture, with 130 trials per method. The results show that client-side proof generation, not network signaling, is the main source of ZKP's latency disadvantage. JWT' s mean call- setup time was 397. 397.737 ms (P 95 = 655. 655.535 ms), compared with 1172.492. 492 ms for ZKP (P 95 = 1643.535. 535 ms). This gap was driven almost entirely by the 775. 775.972 ms proof-generation stage, which also accounts for ZKP's greater variability (SD = 397.604 ms vs. 144.747 ms) and its payload of more than 1,100 bytes, compared with JWT's footprint of under 50 bytes. The payload difference stems from Groth 16's G1/G2 elliptic-curve encoding. Both schemes completed all trials without failure and resisted all tested replay and payload-manipulation attacks. These results indicate that Groth 16 can match JWT's resistance to these specific attack vectors while structurally avoiding private-key exposure, but they do not establish equivalent security against untested threats. In practice, JWT remains preferable when call- setup speed is the binding constraint. Groth16 zk-SNARKs are justified when identity-attribute privacy is a first-order requirement, and client devices have sufficient computational headroom, as in telehealth or confidential consultations. These conclusions are limited by the study's localhost, single- hardware testbed, which does not capture real-world network variability or differences among devices. They are also bounded by a threat model covering only replay and payload-manipulation attacks; we did not evaluate side-channel, denial-of-service, or key-compromise threats. Future work should validate the findings on real, non-localhost networks and diverse client hardware, pursue circuit-level and hardware-accelerated optimizations to narrow the latency gap, and broaden

the threat model before drawing general conclusions about Groth16 zk-SNARKs' suitability for production real-time communication systems.

REFERENCES

Albalwy, F. (2026). Zero-Knowlsedge-Based Policy Enforcement for Privacy-Preserving Cross-Institutional Health Data Sharing on Blockchain. *Systems*, 14(4). https://doi.org/10.3390/systems14040385

Amine, O., Baghery, K., Pindado, Z., & Ràfols, C. (2024). Simulation extractable versions of Groth's zk-SNARK revisited. *International Journal of Information Security*, 23(1), 431–445. https://doi.org/10.1007/s10207-023-00750-7

Asiana, K. P., Huwae, R. B., & Jatmika, A. H. (2025). Studi Eksperimen Keamanan Jaringan Wi-Fi Kampus: Analisis Kerentanan terhadap Serangan Evil Twin dan Deauthentication. *Edumatic: Jurnal Pendidikan Informatika*, *9*(2), 629–638. https://doi.org/10.29408/edumatic.v9i2.31822

Chi, P. W., Lu, Y. H., & Guan, A. (2023). A Privacy-Preserving Zero-Knowledge Proof for Blockchain. *IEEE Access*, 11, 85108–85117. https://doi.org/10.1109/ACCESS.2023.3302691

Darmawan, C., Naibaho, J. P. P., & Kweldju, A. de. (2024). Penerapan Metode Vulnerability Assessment untuk Identifikasi Keamanan Website berdasarkan OWASP ID Tahun 2021. *Edumatic: Jurnal Pendidikan Informatika*, 8(1), 272–281. https://doi.org/10.29408/edumatic.v8i1.25834

De Groef, W., Subramanian, D., Johns, M., Piessens, F., & Desmet, L. (2016). Ensuring Endpoint Authenticity in WebRTC Peer-to-Peer Communication. *Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC '16)*, ACM. https://doi.org/10.1145/2851613.2851804

Dwivedi, A. D., Singh, R., Ghosh, U., Mukkamala, R. R., Tolba, A., & Said, O. (2022). Privacy preserving authentication system based on non-interactive zero knowledge proof suitable for Internet of Things. *Journal of Ambient Intelligence and Humanized Computing*, 13(10), 4639–4649. https://doi.org/10.1007/s12652-021-03459-4

El-Hajj, M., & Oude Roelink, B. (2024). Evaluating the Efficiency of zk-SNARK, zk-STARK, and Bulletproof in Real-World Scenarios: A Benchmark Study. Information, 15(8), 463. https://doi.org/10.3390/info15080463

Fansuri, M. F., & Kusrini, K. (2026). Mutual Information-Driven Feature Selection for Efficient DDoS Detection Using Modern Boosting Ensembles. *Edumatic: Jurnal Pendidikan Informatika*, 10(1), 150–159. https://doi.org/10.29408/edumatic.v10i1.34012

Faozi, R. A., Majid, N. W. A., & Widodo, S. (2026). Security Maturity Assessment of Indonesian Android Mobile Banking Apps using MobSF and OWASP. *Edumatic: Jurnal Pendidikan Informatika*, 10(1), 80–89. https://doi.org/10.29408/edumatic.v10i1.33285

Gaba, G. S., Hedabou, M., Kumar, P., Braeken, A., Liyanage, M., & Alazab, M. (2022). Zero knowledge proofs based authenticated key agreement protocol for sustainable healthcare. *Sustainable Cities and Society*, *80*, Artikel 103766. https://doi.org/10.1016/j.scs.2022.103766

Groth, J. (2016). On the size of pairing-based non-interactive arguments. In M. Fischlin & J.-S. Coron (Eds.), *Advances in Cryptology: EUROCRYPT 2016*. Computer Science, Vol. 9666, pp. 305-326). Springer. https://doi.org/10.1007/978-3-662-49896-5_11

Kuznetsov, O., Kuznetsova, Y., Ziyatbekova, G., Kovalenko, Y., & Palahusynets, R. (2026). Performance Evaluation of zk-SNARK Protocols for Privacy-Preserving Sensor Data Verification: A Systematic Benchmarking Study. *Sensors*, *26*(8), 2486. https://doi.org/10.3390/s26082486

Li, W., Meese, C., Zhong, Z. G., Guo, H., & Nejad, M. (2021). Location-aware verification for autonomous truck platooning based on blockchain and zero-knowledge proof. *IEEE International Conference on Blockchain and Cryptocurrency (ICBC 2021)*. https://doi.org/10.1109/ICBC51069.2021.9461116

Mahmoud, H., & Abozariba, R. (2025). A systematic review on WebRTC for potential applications and challenges beyond audio video streaming. *Multimedia Tools and Applications*, 84, 2909–2946. https://doi.org/10.1007/s11042-024-20448-9

Maruschke, M., Zimmermann, J., Bach, T., & Haensge, K. (2015). The role of QoS in WebRTC and IMS-based IPTV services. *International Journal on Advances in Telecommunications*, *8*(1 & 2), 59–73.

Oude Roelink, B., El-Hajj, M., & Sarmah, D. (2024). Systematic review: Comparing zk-SNARK, zk-STARK, and bulletproof protocols for privacy-preserving authentication. *Security and Privacy*, 7(5). https://doi.org/10.1002/spy2.401

Putra, N. A., Sidiq, M. F., & Amrulloh, A. (2025). Enhanced Data Security in Video Media using RSA-LSB Hybrid Technique. *Edumatic: Jurnal Pendidikan Informatika*, 9(2), 551–559. https://doi.org/10.29408/edumatic.v9i2.31030

Salleras, X., & Daza, V. (2020). SANS: Self-sovereign authentication for network slices. *Security and Communication Networks*, 2020, 8823573. https://doi.org/10.1155/2020/8823573

Sassi, Z. ben, Yaici, C. C., Xiang, J., Salem, O., & Mehaoua, A. (2025). Benchmarking Zero-Knowledge Proof-Based Authentication Protocols. *Proceedings of 2025 IEEE International Symposium on Future Telecommunication Technologies, SOFTT 2025*, 131–137. https://doi.org/10.1109/SOFTT67007.2025.11212915

Skowroński, R. (2026). WebRTC Swarms: Decentralized, Incentivized, and Privacy-Preserving Signaling with Designated Verifier Zero-Knowledge Authentication. *Future Internet*, 18(1). https://doi.org/10.3390/fi18010013

Su, Z., Wang, S., Cai, H., Huang, J., Chen, Y., Zhang, X., & Alam, M. (2024). A Secure and Efficient Authentication Scheme for Large-Scale IoT Devices Based on Zero-Knowledge Proof. *Electronics (Switzerland)*, 13(18). https://doi.org/10.3390/electronics13183735

Zhao, R. (2024). Toward the flow-centric detection of browser fingerprinting. Computers & Security, 137, 103642. https://doi.org/10.1016/j.cose.2023.103642