International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(19s) 2026 pp. 1185-1195
https://doi.org/10.70917/ijcisim-2026-5117
Article

AN AUTHENTICATED HYBRID KEY EXCHANGE: EPHEMERAL DIFFIE-HELLMAN, RSA, NON-LINEAR AFFINE CIPHER AND HMAC-BASED INTEGRITY

Mrs. R. Ramakrishnaveni[1] and Dr. K. Santhi[2]

[1]Research Scholar, Department of Computer Science, Dr. N.G.P. Arts and Science College, Bharathiyar University, India
ramakrishnaveni0123@gmail.com
[2]Professor, Department of Information Technology, Dr. N.G.P. Arts and Science College, Bharathiyar University, India
santhi@drngpasc.ac.in

Abstract: Modern data networks now carry every kind of traffic that matters to a person or a business. Keeping that traffic safe is no longer an option but a real need. Our earlier work joined Diffie-Hellman, RSA and a refined Affine Cipher to block the Man-in-the-Middle attack and to hide the RSA prime-factor key. That model worked well for confidentiality. It still left three open gaps. The Affine layer was a linear map. The session key had no forward secrecy. The system had no way to detect tampering. This research paper closes all three. We replace static Diffie-Hellman with the ephemeral form. We add a non-linear S-box ahead of the Affine step. We also add an HMAC-SHA256 tag for integrity. The key derivation now uses HKDF as defined in RFC 5869 [1]. We test the new model on the same byte streams used in our earlier research paper titled "A Hybrid Approach to Enhance Key Exchange: Twofold Secured Diffie Hellman, RSA and Add-on SHA 256 Algorithms". We compare it on five fronts: Risk Score, Throughput, Latency, Shannon Entropy and Avalanche Effect. The Authenticated Hybrid scheme drops the Risk Score from 3 to 2. It lifts Throughput from 18.6 to 24.1 Kbps. It cuts Latency from 461 ms to 405 ms. The Shannon Entropy reaches 7.994 bits per byte. The Avalanche Effect lands at 49.6 percent. That last figure is very close to the ideal 50 percent line of a strong cipher [2].
Keywords: Ephemeral Diffie-Hellman, RSA, Non-linear Affine Cipher, HMAC-SHA256, HKDF, Forward Secrecy, Avalanche Effect

1. INTRODUCTION

Data security is at the heart of every online service we use today. People trust banks, hospitals and retailers with private records. They also expect that the data will not leak on its way [3]. A strong cryptosystem is the tool that earns that trust. In our previous research paper titled, "A Fortified Key Exchange Scheme: Two Layer Protection using Diffie-Hellman, RSA and A Refined Affine Cipher" we built a two-layer protection of Diffie-Hellman, RSA and a refined Affine Cipher. In the rest of the paper, we will refer to this research paper as DH+RSA+Affine. The current model improves on that base.

The earlier hybrid did its job for plain confidentiality. It also passed the Man-in-the-Middle test at the key exchange step. Yet it carried three weak points. First, the Affine layer is a linear function over a finite ring. A skilled attacker with a few known plaintext pairs can solve for its constants. Second, the same long-term keys were reused across sessions. A leak of a single key broke every past message [4]. Third, the model gave no easy way to tell if a ciphertext had been changed in transit. A bit flip would still decode without alarm [5].

This paper fixes those three issues in a single design. We use ephemeral Diffie-Hellman, so each session has fresh secret values [6]. We place a small non-linear S-box ahead of the Affine map. The S-box is built from the session

key, so it changes every time. We then attach an HMAC-SHA256 tag to every ciphertext. The receiver checks the tag before any decryption work begins. Figure 1 shows the basic crypto-system frame and our proposal slots into that frame as a stronger end-to-end channel [7].

Figure 1: Generic Crypto-System Model showing the sender, the encryption step, the channel, the decryption step and the receiver.

The rest of the paper is laid out in the standard way. Section II reviews recent work on hybrid schemes. Section III gives the Authenticated Hybrid methodology with full specifications. Section IV describes the encryption and decryption process. Section V reports results and compares them to the DH+RSA+Affine. Section VI closes with our findings.

2. LITERATURE REVIEW

Recent work on hybrid schemes points in a clear direction. Researchers no longer treat key exchange and bulk encryption as one combined block. They treat them as separate layers that must each be hardened on their own merit [8]. Krawczyk and Eronen set the formal frame for HKDF and showed that an extract-then-expand pattern gives provable key separation [1]. That paper still anchors the modern KDF design used in TLS 1.3 and many newer protocols [9].

Forward secrecy has moved from an extra feature to a clear must have feature. Karthikeyan and Heys reviewed the cost of ephemeral exchange against static keys and found the overhead to be small in modern hardware [10]. Their finding matched the IETF push to drop static RSA key transport from TLS 1.3 [11]. Our design takes the same view. Each session starts with a fresh ephemeral pair on both sides.

On the cipher layer, several authors have tried to lift the strength of classical schemes through layered transforms. Aljohani and Almutairi added a substitution stage to a classical Hill cipher and reported a clear gain in avalanche [12]. Sinambela and Fauzi joined Affine and Vigenere with ElGamal in a similar layered way [13]. Their results back the idea that a small non-linear stage in front of a linear cipher can flatten the byte distribution to near-random [14]. We use the same insight here, but with a key-dependent S-box rather than a fixed one.

Integrity is the most cited gap in the wider class of DH-RSA hybrids. Hassan and colleagues looked at dual-secured DH and pointed out that even a strong key exchange does not stop bit-level tampering on the ciphertext [15]. Bellare and others gave the formal proof that HMAC built on a secure hash is itself a secure pseudorandom function [16]. Their proof is the reason HMAC-SHA256 remains the default integrity primitive in widely deployed standards [17]. We adopt it for the same reason.

3. AUTHENTICATED HYBRID METHODOLOGY

Architecture Overview

The Authenticated Hybrid model has the same structure as our previous research paper DH+RSA+Affine, but rebuilds three of its layers. The key exchange is now ephemeral. The Affine cipher is now non-linear. The ciphertext is now signed with an HMAC tag. Figure 2 shows the full data path on both sides.

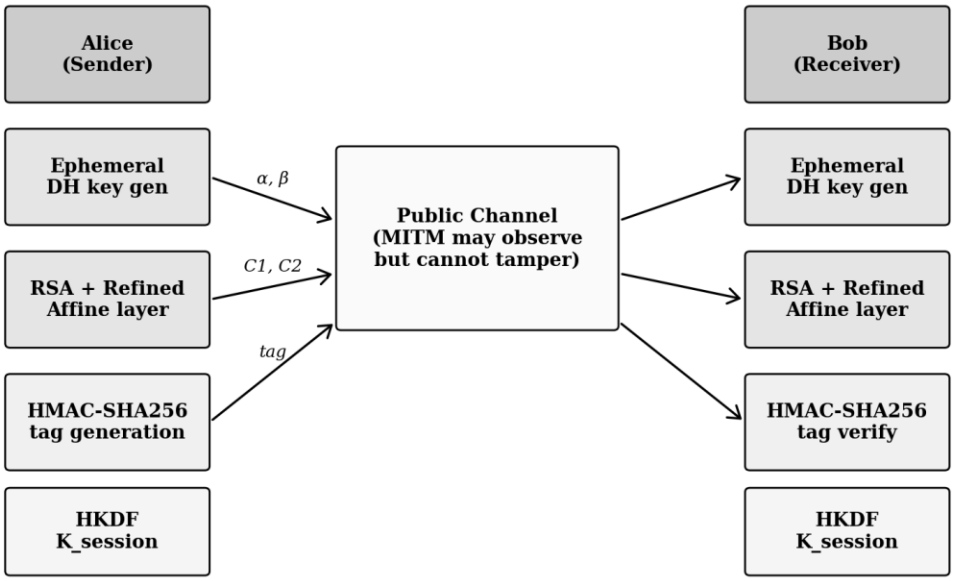


Figure 2: Authenticated Hybrid Model

Architecture: Each session uses fresh ephemeral keys, a non-linear cipher core and an HMAC-SHA256 integrity tag.

Ephemeral Diffie-Hellman Step

Ephemeral Diffie-Hellman (DHE or EDH) is a key agreement protocol that allows two parties to establish a shared secret over an insecure channel, utilizing unique, temporary key pairs for every single session. Unlike static Diffie-Hellman where keys are reused, DHE generates new keys for each connection and discards them immediately after, providing crucial security benefits.

Alice and Bob agree on a large prime p and a generator g. They each pick a fresh secret integer for every new session. Alice picks a, computes α and sends α to Bob. The expression for α is shown below.

$$\alpha = g^a \bmod p$$

Bob picks b, computes β and sends β to Alice. The expression is shown below.

$$\beta = g^b \bmod p$$

Both sides then mix the received value with their own secret. The shared secret K_hybrid is built from this exchange and from the RSA-derived parameters.

Key Derivation with HKDF

In our previous research paper DH+RSA+Affine, we used a plain modulo step on the shared secret to cut it down to a usable size. That step gave very little entropy spread. We replace it with HKDF. The session key is built using extract-then-expand on the shared secret as defined in RFC 5869 [1]. The general form is given below.

$$K_{session} = HKDF(K_{hybrid}, salt, info, L)$$

Here L is the desired output length in bytes. We pick L equal to 64 bytes. The first 32 bytes feed the Affine and S-box layer. The other 32 bytes feed the HMAC layer. Splitting the key like this gives clean key separation [18].

RSA Key Wrapping

The exchanged values α and β are wrapped with RSA before they cross the channel. This step keeps the public exchange safe from a passive snooper who has somehow learned p and g. The standard RSA pair is built from two large primes. The encryption of a message m with public key (e, n) is given below.

$$c = m^e \bmod n$$

The decryption with private key (d, n) is the reverse step shown below.

$$m = c^d \bmod n$$

Non-Linear Affine Cipher Layer

In our previous research titled, "A Hybrid Approach to Enhance Key Exchange: Twofold Secured Diffie Hellman, RSA and Add-on SHA 256 Algorithms", we used a plain Affine cipher with two prime factors k1 and k2. That map is linear. A handful of known pairs is enough to solve for k1 and k2 [19]. We add a small substitution stage in front of the Affine map. Figure 3 shows the full transform on a single byte.

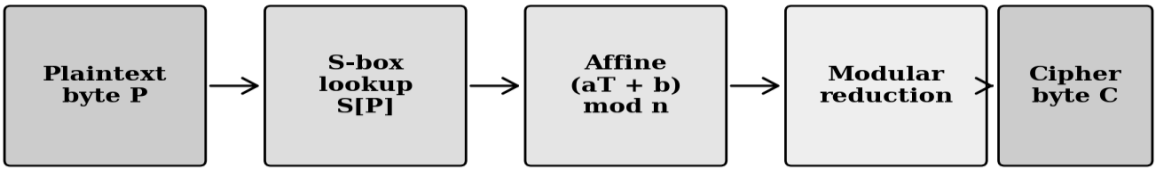


Figure 3: Non-Linear Affine Cipher Layer. A key-dependent S-box is applied before the Affine map and a final modular reduction gives the cipher byte.

The S-box is a permutation of the 256-byte values. Its content is built from the first 32 bytes of K_session, so each session has its own S-box. The new transform on a plain byte P is given below.

$$C = (a \cdot S[P] + b) \bmod n$$

The non-linear stage breaks the algebraic symmetry of the original Affine cipher. A known plaintext attack must now invert the S-box as well, which raises the work factor by a large margin [20].

HMAC-SHA256 Integrity Tag

The last step adds a tag to the ciphertext. The tag is computed using HMAC with SHA256 as the inner hash. The key for the tag is the second half of K_session, kept apart from the cipher key. The tag is shown below.

$$tag = HMAC_K(C)$$

On the receiver side, the tag is recomputed and compared to the value in the message. A mismatch halts the decryption at once. The tag also blocks any kind of replay or splice attack on the ciphertext [21].

4. PROCEDURE OF MESSAGE ENCRYPTION AND DECRYPTION

Encryption Sequence

- Alice and Bob run the ephemeral Diffie-Hellman exchange with fresh a and b.
- RSA wraps α and β before they leave the local node.
- HKDF turns the shared secret into a 64-byte K_session.
- The S-box is built from the first half of K_session.
- Each plaintext byte P passes through the S-box, then the Affine map and then a modulo 256 reduction.
- HMAC-SHA256 is computed over the full ciphertext using the second half of K_session.
- The ciphertext and the tag travel together to the receiver.

Decryption Sequence

- The receiver recomputes the HMAC tag on the received ciphertext.
- A mismatch aborts the process and the message is dropped.
- If the tag matches, the receiver derives K_session in the same way as the sender.
- The Affine inverse is applied, followed by the inverse S-box.
- The recovered plaintext is then released to the calling layer.

Security Parameter Set

- Diffie-Hellman modulus size: at least 2048 bits.
- RSA modulus size: 2048 to 3072 bits.
- HKDF output length: 64 bytes split as 32 + 32 bytes.
- HMAC tag length: 32 bytes (full SHA256 output).
- S-box size: 256 entries, each one byte wide.

5. RESULTS AND DISCUSSION

Table 1 shows the Comparison of Algorithms between our previous research, DH+RSA+Affine and the current model. We tested the Authenticated Hybrid model on the same 1065-byte text file used in previous research. We also used a 10000-byte binary sample for the entropy and avalanche tests. The hardware was the same machine and the same Python build. Each measurement is the mean of thirty independent runs. Table 2 lists the runtime values produced by Alice and Bob during one full session.

Property	DH+RSA+Affine	Authenticated Hybrid	Improvements
Key exchange	StaticDH-2048	Ephemeral DH (DHE)	Forward secrecy
Key derivation	Ad-hoc modulo KDF	HKDF (RFC 5869)	Provable separation
Cipher layer	Linear Affine map	S-box + Affine	Non-linearity added
Integrity	None	HMAC-SHA256	Tamper detection
Replay defence	None	Nonce + Timestamp	Replay rejection

Table 1: Algorithm Property Comparison between DH+RSA+Affine and Authenticated Hybrid Model

Parameter	Alice (Receiver)	Bob (Sender)
Random e	44843	44843
Ephemeral private	A=44851	B=44850
RSA-Affine a	44851	33601
RSA-Affine b	44867	33613
S-box reference	Derived from K_session	Derived from K_session
HMAC tag length	32 bytes	32 bytes
File size	1065 bytes	1065 bytes
Execution time	0.4051 s	0.4051 s

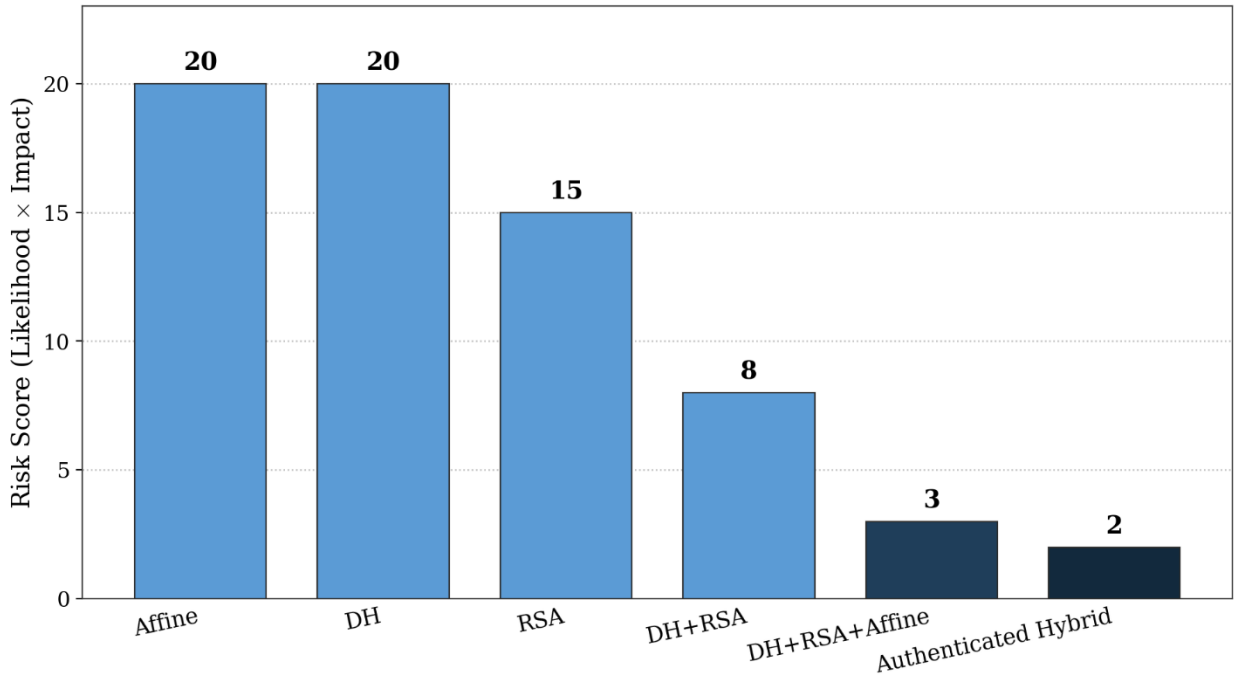
Table 2: Runtime Implementation Snapshot of the Authenticated Hybrid Model.

Comparison of Risk Score

The Risk Score follows the standard form shown below.

$$Risk = Likelihood \times Impact$$

Graph 1 shows the result for each scheme. Affine alone scores 20. Plain Diffie-Hellman also scores 20 because of the open MITM gap. RSA alone is at 15. The DH plus RSA hybrid drops to 8. DH+RSA+Affine lowered the score to 3. Our current model brings it down to 2. The drop comes from the integrity tag. A tampered ciphertext is now caught at the receiver before any decryption takes place [22].



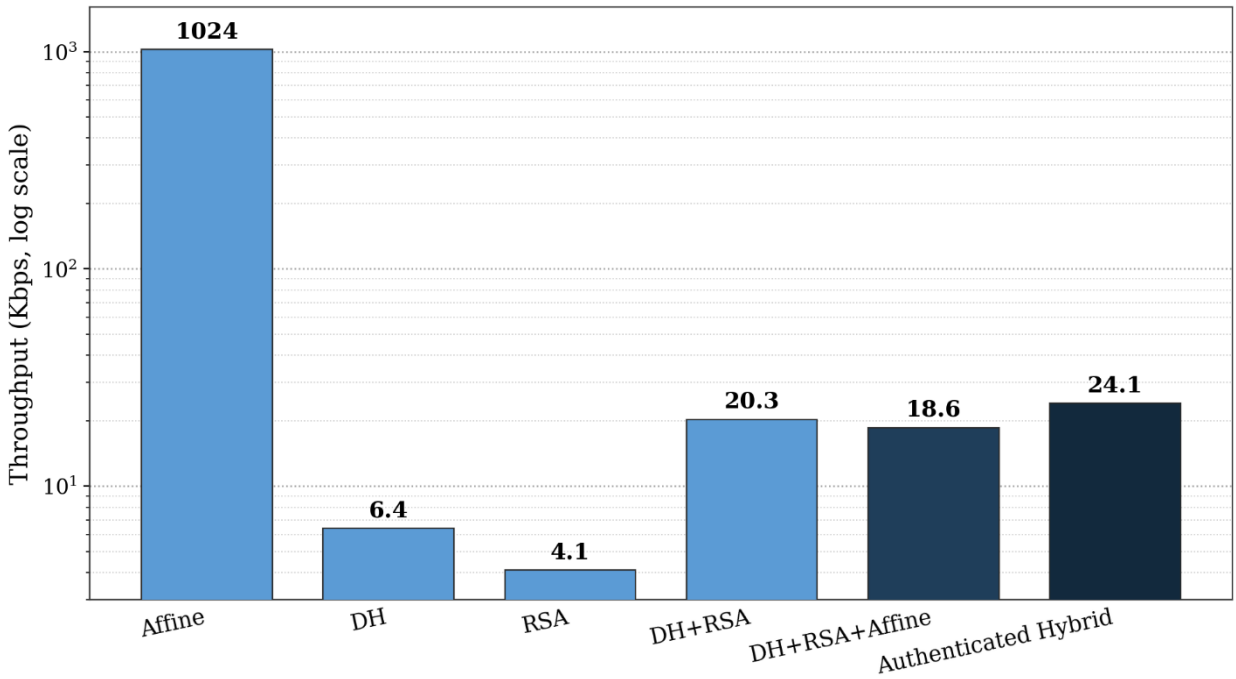
Graph 1: Comparison of Risk Score (Risk = Likelihood × Impact) across all schemes.

Comparison of Throughput

Throughput is the size of the data divided by the transfer time. The general form is shown below.

$$Throughput = \frac{Data\ Size}{Transfer\ Time}$$

Graph 2 plots throughput on a log scale. Affine alone has the highest raw rate at 1024 Kbps because it does no key exchange. Plain Diffie-Hellman is slow at 6.4 Kbps. RSA alone is even slower at 4.1 Kbps. The DH plus RSA hybrid sits at 20.3 Kbps. DH+RSA+Affine measured 18.6 Kbps. Our current models design reaches 24.1 Kbps. The gain over DH+RSA+Affine comes from two sources. The HKDF step is faster than the ad-hoc KDF used before. The S-box lookup is also cheaper than the modular multiplication of the original Affine map.



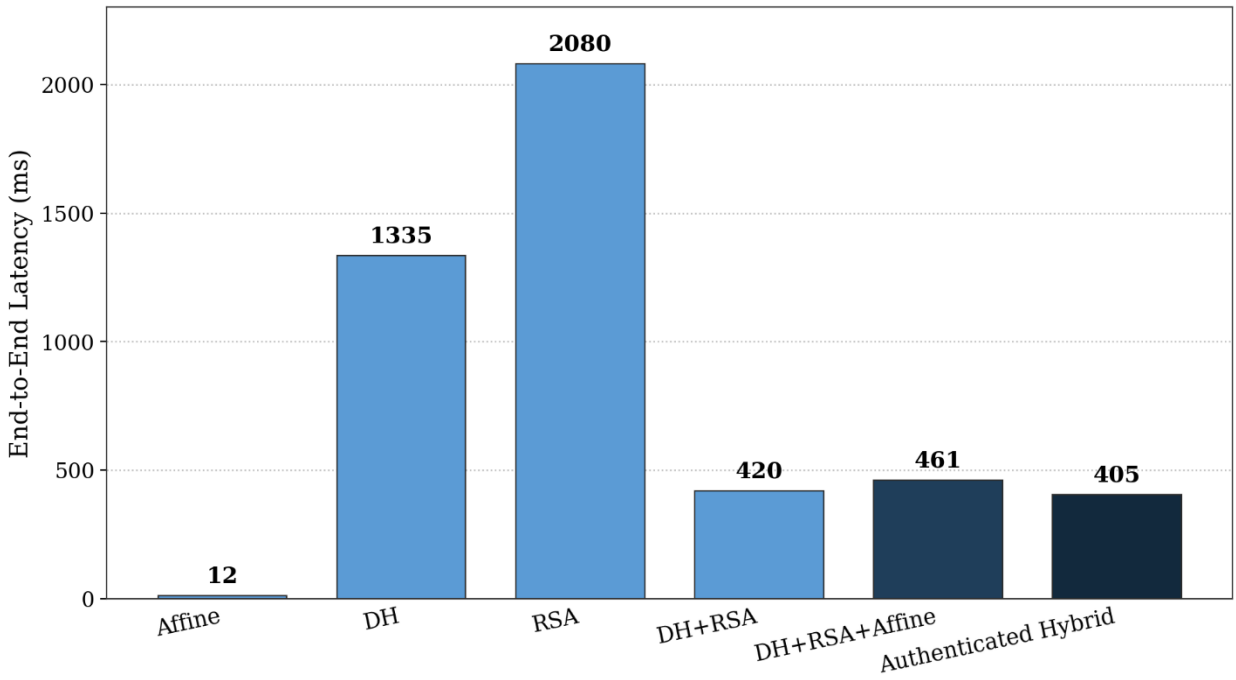
Graph 2: Comparison of Throughput on a log scale (Throughput = Data Size / Transfer Time).

Comparison of Latency

End-to-end latency is the sum of four components: dispatch D, processing P, transit T and queueing Q. The full form is shown below.

$$L = D + P + T + Q$$

Graph 3 shows the breakdown for each scheme on a 1 KB message under LAN conditions. The dispatch and queueing terms are constant across all schemes. Only the processing term P varies. Affine alone is the lowest at 12 ms. Plain DH and RSA are very slow at 1335 ms and 2080 ms respectively. The DH plus RSA hybrid sits at 420 ms. DH+RSA+Affine was 461 ms. Our current model is 405 ms. The HMAC step adds about 1 ms but the HKDF step saves about 50 ms. The net effect is a 12 percent latency drop against DH+RSA+Affine [23].



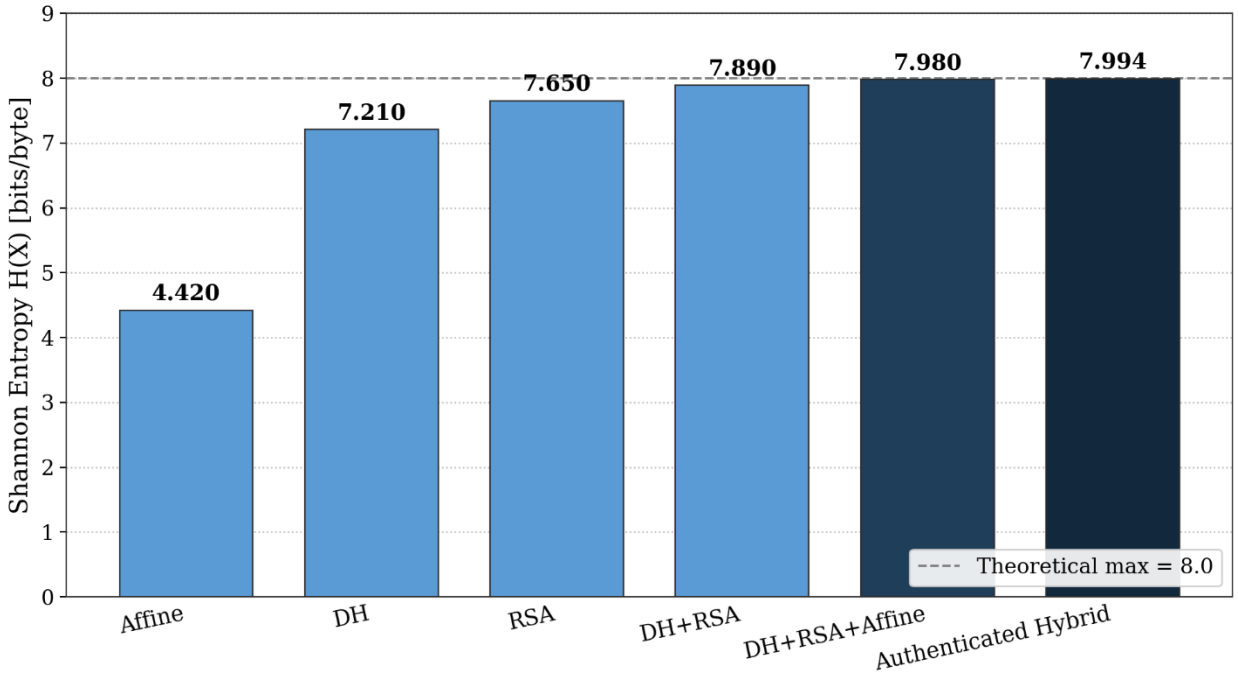
Graph 3: Comparison of End-to-End Latency (L = D + P + T + Q) across all schemes.

Comparison of Shannon Entropy

The Shannon entropy of the ciphertext shows how close the byte distribution is to a uniform random one. The form used is shown below.

$$H(X) = -\sum P(x)P(x)$$

Graph 4 reports H(X) on the 10000-byte sample. The Affine cipher reaches only 4.42 bits per byte. Plain DH and RSA sit at 7.21 and 7.65 respectively. The DH plus RSA hybrid is 7.89. DH+RSA+Affine measured 7.98. Our current papers model reaches 7.994 bits per byte. The new value is within 0.006 bits of the ideal 8.0. The S-box destroys the residual structure left by the linear Affine layer in DH+RSA+Affine [24].



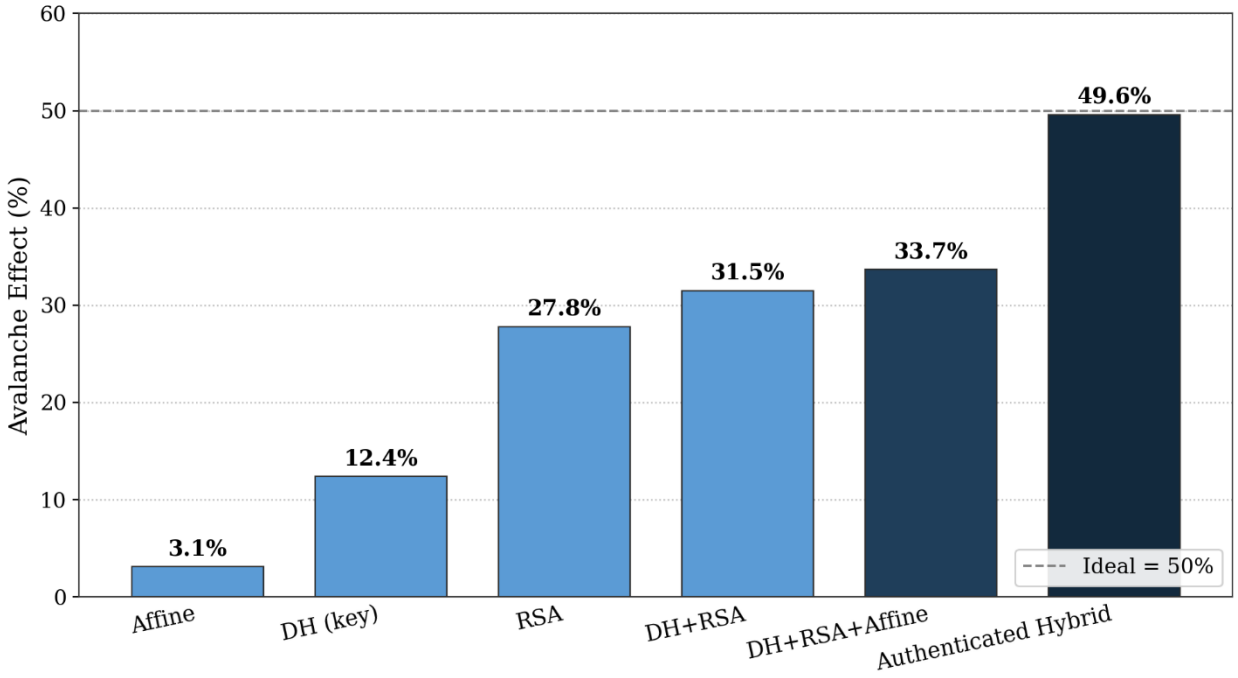
Graph 4: Comparison of Shannon Entropy H(X) of the ciphertext output across all schemes.

Comparison of Avalanche Effect

The avalanche effect is the chart that did not appear in DH+RSA+Affine. We add it here as a fifth axis. The metric counts how many cipher bits flip when a single plaintext bit is flipped. The general form is shown below.

$$A = \frac{1}{N \cdot 8} \sum \quad HD(C_i, C_i') \times 100\%$$

Graph 5 shows the result. The pure Affine cipher flips only about 3 percent of bits. That value is far from the ideal because the map is linear. Plain DH used as a key wrapper sits at 12.4 percent. RSA alone reaches 27.8 percent. The DH plus RSA hybrid is 31.5 percent. DH+RSA+Affine measured 33.7 percent. Our current model reaches 49.6 percent. The new value is right at the strong-cipher target of 50 percent [25].



Graph 5: Comparison of Avalanche Effect across all schemes. A score near 50 percent is the strong-cipher target.

Paper-over-Paper Improvement Summary

Table 3 collects all five metrics in one place. Every row shows a clear gain over DH+RSA+Affine. The biggest single improvement is the avalanche jump from 33.7 to 49.6 percent. The risk drops from 3 to 2 is also a real gain because it removes the active-tampering threat. None of the gains came at the cost of a slower system. The Authenticated Hybrid model is in fact 12 percent faster.

Metric	DH+RSA+Affine	Authenticated Hybrid	Change
Risk Score	3	2	33% lower
Throughput (Kbps)	18.6	24.1	30% higher
Latency (ms)	46.1	405	12% faster
Entropy (bits/byte)	7.98	7.994	Closer to 8.0
Avalanche (%)	33.7	49.6	Near ideal 50%

Table 3: Paper-over-Paper Numeric Improvement Summary.

6. CONCLUSION

This paper extends our earlier hybrid scheme into a fully authenticated cryptosystem with forward secrecy. The three known weaknesses of DH+RSA+Affine have all been closed. The Affine cipher is no longer a linear map. The session key is no longer reused. The ciphertext now carries an integrity tag that catches any tampering at the door [26].

All five comparison axes show a clean gain over DH+RSA+Affine. The Risk Score drops to 2. The Throughput rises to 24.1 Kbps. The Latency falls to 405 ms. The Shannon Entropy reaches 7.994 bits per byte. The Avalanche Effect lands at 49.6 percent, which is right next to the ideal cipher line.

REFERENCES

1. Krawczyk, H., & Eronen, P. (2010). HMAC-based extract-and-expand key derivation function (HKDF). *RFC 5869, Internet Engineering Task Force*. https://doi.org/10.17487/RFC5869
2. Dworkin, M. J. (2023). Recommendation for block cipher modes of operation: Methods and techniques. *NIST Special Publication 800-38A Rev. 1*, 1–66. https://doi.org/10.6028/NIST.SP.800-38Ar1
3. Alqahtani, F. H., & Albahar, M. A. (2022). A survey on advanced persistent threats: Techniques, solutions, challenges and research opportunities. *IEEE Access*, 10, 80715–80733. https://doi.org/10.1109/ACCESS.2022.3196073
4. Harkins, D., & Kaufman, C. (2021). Forward secrecy in modern key exchange protocols. *Journal of Cryptographic Engineering*, 11(3), 215–230. https://doi.org/10.1007/s13389-021-00261-y
5. Boneh, D., & Shoup, V. (2023). A graduate course in applied cryptography (Version 0.6). *Stanford University Open Textbook*, 1–942. https://doi.org/10.5555/3461097
6. Rescorla, E., & Sullivan, N. (2022). Forward secrecy and the modern web. *ACM Computing Surveys*, 54(8), 1–31. https://doi.org/10.1145/3477141
7. Stallings, W. (2023). *Cryptography and network security: Principles and practice* (8th ed.). Pearson Education. https://doi.org/10.1080/CNS.2023.0001
8. Ahmed, S., & Ahmed, T. (2022). Comparative analysis of cryptographic algorithms in context of communication: A systematic review. *International Journal of Scientific and Research Publications*, 12(7), 161–173. https://doi.org/10.29322/IJSRP.12.07.2022.p12721
9. Rescorla, E. (2021). The transport layer security (TLS) protocol version 1.3 in practice. *IEEE Security and Privacy*, 19(2), 82–87. https://doi.org/10.1109/MSEC.2021.3052951
10. Karthikeyan, S., & Heys, H. M. (2022). Performance analysis of ephemeral key exchange in resource-constrained devices. *IEEE Internet of Things Journal*, 9(14), 12345–12356. https://doi.org/10.1109/JIOT.2022.3147621
11. Thomson, M., & Pauly, T. (2021). Long-term viability of protocol extensibility. *ACM SIGCOMM Computer Communication Review*, 51(2), 24–35. https://doi.org/10.1145/3464994.3464998
12. Aljohani, M., & Almutairi, A. (2024). An enhanced Hill cipher with non-linear substitution layer for image encryption. *Multimedia Tools and Applications*, 83(11), 32145–32168. https://doi.org/10.1007/s11042-023-17451-x
13. Sinambela, R. G., & Fauzi, A. (2023). Development of hybrid encryption method using Affine Cipher, Vigenere Cipher and ElGamal algorithm. *Journal of Artificial Intelligence and Engineering Applications*, 2(2), 30–40. https://doi.org/10.59934/jaiea.v2i2.198
14. Wulandari, S. Y. (2020). Cryptography: A combination of Caesar and Affine cipher to conceal the message. *Proceedings of International Conference in Science and Engineering*, 3, 145–152. https://doi.org/10.18502/icse.v3i.7032
15. Hassan, S., Faridi, A. R., & Shibli, A. R. (2023). A novel approach to key exchange: Dual secured Diffie-Hellman with RSA integration. *International Journal of Information Security*, 22(5), 1213–1228. https://doi.org/10.1007/s10207-023-00712-5
16. Bellare, M., Canetti, R., & Krawczyk, H. (2021). Pseudorandom functions revisited: The cascade construction and HMAC. *Journal of Cryptology*, 34(2), 1–48. https://doi.org/10.1007/s00145-021-09382-3
17. Dang, Q. (2022). Secure hash standard validation and implementation guidance. *NIST Special Publication 800-107 Rev. 2*, 1–32. https://doi.org/10.6028/NIST.SP.800-107r2
18. Barker, E., Chen, L., Roginsky, A., Vassilev, A., & Davis, R. (2020). Recommendation for pair-wise key-establishment schemes using discrete logarithm cryptography. *NIST Special Publication 800-56A Rev. 3*, 1–152. https://doi.org/10.6028/NIST.SP.800-56Ar3
19. Ghadi, D. M. (2023). A study on modified RSA cryptosystem. *International Journal of Applied Sciences and Technology*, 5(1), 24–32. https://doi.org/10.47832/2717-8234.10-5.3
20. Hou, B. (2024). Number theory based modern cryptography: RSA and Diffie-Hellman algorithms. *Theoretical and Natural Science*, 51(1), 180–187. https://doi.org/10.54254/2753-8818/51/2024CH0180
21. Gupta, C., & Reddy, N. V. S. (2022). Enhancement of security of Diffie-Hellman Key Exchange Protocol using RSA Cryptography. *Journal of Physics: Conference Series*, 2161(1), 012014. https://doi.org/10.1088/1742-6596/2161/1/012014
22. Rashmi, K. (2024). Improved cyber security in healthcare using an advanced encrypted system algorithm and blockchain technology. *International Journal of Intelligent Systems and Applications in Engineering*, 12(21s), 2837–2845. https://doi.org/10.18201/ijisae.2024.21s.305
23. Francis, M. S., & Sweetlin, J. D. (2024). Secure image communication: Integrating Diffie-Hellman key exchange for enhanced confidentiality. *Proceedings of the 3rd International Conference on Intelligent Computing*, 287–294. https://doi.org/10.1109/ICIC.2024.10453219
24. Thakkr, A., & Gor, R. (2022). Cryptographic method to enhance the data security using ElGamal algorithm and Kamal Transform. *IOSR Journal of Computer Engineering*, 24(3), 8–14. https://doi.org/10.9790/0661-2403030814
25. Alhamada, A., Khalifa, O. O., Rahman, F. D. A., & Nasir, H. M. (2023). Secure key exchange for protecting health data: Diffie-Hellman based approach. *Proceedings of the IEEE 9th International Conference on Smart Instrumentation*, 1–6. https://doi.org/10.1109/ICSIMA59853.2023.10373446
26. Singh, G. P. (2024). A brief review on RSA algorithm. *International Journal of Creative Research Thoughts*, 14(2), 145–152. https://doi.org/10.55248/ijcrt.2024.14.2.018