Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
DOI : 10.47974/JIOS-2239

End-to-end encryption architecture for enhancing confidentiality and integrity in secure online examination system

Arif Khan [†]
Siddhartha Sankar Biswas *
Bhavya Alankar [§]
Harleen Kaur [‡]
Sherin Zafar [@]
Department of Computer Science & Engineering
School of Engineering Sciences & Technology (SEST)
Jamia Hamdard (Deemed to be University)
Hamdard Nagar
New Delhi 110062
India

Abstract

During online exams, the primary focus is usually on using secure, reliable & non-duplicative communication methods. In contrast, many individuals try to access exam Questions and/or alter the answers given by students who take those exams. In addition to people trying to access examination materials, there are also a number of people who may try and use someone else's identity to take that exam. The purpose of the proposed architecture discussed in this paper is to provide an end-to-end method of encryption utilizing particle swam optimization (PSO), ChaCha20 such that it can be used to provide better ways of generating keys, producing more randomness with keystreams, and reducing latency time incurred due to encryption during the entire process of taking the exam, from the time that question papers are issued until the students submit their answers. To accomplish this, although the primary means of validating the integrity of the entire exam process was SHA-256, a time-based one-time password was used to verify the user's identity. Based on experimental results, the proposed architecture demonstrated significant reductions in both encryption/decryption times up to 24% as compared to current encryption methods like AES-256 and RSA-1024, and achieved higher throughput than other existing encryption methods. This indicates that the proposed architecture has a high level of resistance to a range of attack types, including statistical, differential, and man-in-the-middle attacks,

[†] *E-mail:* **arif.sabirkhan@gmail.com**
* *E-mail:* **ssbiswas@jamiahamdard.ac.in (Corresponding Author)**
[§] *E-mail:* **balankar@jamiahamdard.ac.in**
[‡] *E-mail:* **harleen@jamiahamdard.ac.in**
[@] *E-mail:* **sherin.zafar@jamiahamdard.ac.in**

© Taru Publications

and extremely low processing overhead, making it well-suited for large-scale, real-time examination environments.

Subject Classification (2020): *68P25, 94A60, 94A62, 68M10.*

Keywords: *ChaCha20, End-to-end encryption, Integrity verification, Online examination systems, Particle swarm optimization, Security architecture.*

1. Introduction

There are many forms of online testing, but many students have concerns about the security of their information, particularly regarding the privacy of their personal data, data integrity, and identity verification. Studies and research indicate that misuse of students' responses and the ability to change what students do during testing continue to pose a risk, particularly with online assessments, when students' data has travelled across many systems/clouds/ third-party servers [1]. This presents a significant issue: server-side encryption fails to safeguard critical test data during its transit to or through external services. End-to-end encryption (E2EE) addresses this issue by ensuring that only the candidate and the exam controller can access unencrypted data [2].

ChaCha20 is a novel stream cipher that outperforms AES in terms of efficiency, power consumption, and security when utilized with software. SHA-256 remains a popular choice in many secure communication applications, blockchain methods, and digital forensics due to its inherent strength against compromise and relative simplicity. One of the standard methods to protect valuable software systems from user credential seizure by hackers, credential reuse, and perpetrators masquerading as test takers is to utilize multifactor authentication, such as a one-time password (OTP) [3]. Researchers have applied metaheuristic optimization algorithms, specifically Particle Swarm Optimization (PSO), to maximize encryption strength. For instance, PSO can be used to implement ChaCha20 parameters that make the keystream more dissimilar and stochastic.

This paper proposes a robust implementation of a comprehensive encryption system for online tests that is independent of third-party services. The novel contribution of the proposed work is as follows:

1. ChaCha20 cipher optimized for lightweight data privacy through PSO.
2. SHA-256 hashing for complete integrity verification.
3. OTP based authentication for validation of candidate identity.

This paper is organized as follows: the related work is presented in Section 2, while the System model is discussed in Section 3. The proposed work is discussed in Section 4, followed by Section 5, which discusses the results and discussion. Finally, Section 6 discusses the Conclusion and Future direction.

2. Related Work

Initial planning for secure online exams focused on proctoring and cheating prevention rather than cryptographic hardening of the transmission channel. Langenfeld [4] and Mutongoza and Olawale [5] noted that, while there was a rush to move to online examinations, serious risks remained in remote assessment workflows, including impersonation and data leakage. Most traditional online assessment platforms encrypt communication using HTTPS/TLS, but TLS-only provides point-to-point protection. ChaCha20 performs better than AES; in environments without hardware acceleration, it exhibits low computational overhead while maintaining strong security guarantees [6]. Recently, ChaCha20 has demonstrated efficiency in constrained-resource environments, such as the Internet of Things (IoT), mobile devices, and embedded devices [7]. SHA-256 remains one of the most widely used hash algorithms, characterized by strong security properties and sufficient computational feasibility. The work by Encryption Consulting discusses SHA-256 and its suitability for digital forensics, integrity checking, and secure communication protocols. Multi-factor mechanisms, especially OTPs, provide enhanced protection and are widely used in online banking and educational settings [8]. Nevertheless, none of the previous works utilized PSO to optimize ChaCha20 in online examination systems or combined PSO-optimized ChaCha20 with an end-to-end encrypted architecture.

1. No inclusion of PSO-optimized ChaCha20 encryption for exam data in terms of security and performance.
2. No full end-to-end framework that takes secure encryption, integrity verification (SHA-256), and OTP-based authentication into consideration.
3. Limited studies that analyze encryption latency and integrity verification overhead, and the impact of real-time for an exam environment.

These limitations underscore the need to develop the proposed PSO–ChaCha20–SHA256–OTP-based E2EE architecture outlined in the following section.

3. System Model, Problem Formulation, and Threat Model

This section formalizes the communication aspects of the envisioned online examination system, characterizes the major security concerns we identified, and specifies the threat model we used throughout this paper. Figure 1 illustrates the overall interaction between trusted endpoints and untrusted intermediary servers, as well as adversarial capabilities.

3.1 *System Model*

The online examination environment is modeled as a tuple $S = (U, E, N, K, H, A_{\mathrm{OTP}})$, where:

- U: Candidate device (trusted endpoint).
- E: Exam controller/server (trusted endpoint).

- $N = \{n_1, n_2, \ldots, n_m\}$ Intermediary servers (cloud routers, CDNs, gateways), all untrusted.
- K: Key space of ChaCha20 (256-bit), optimized using PSO.
- H: SHA-256 integrity verification function.
- A_{OTP}: OTP-based authentication mechanism.

Let

- Q: Question paper or exam instructions.
- R: Candidate response data.
- C_Q and C_R: ChaCha20-encrypted versions of Q and R.
- $h_Q = SHA256(C_Q)$, $h_R = SHA256(C_R)$ Integrity hashes.

All data transmissions occur over an insecure network $U \xrightarrow{C} N \xrightarrow{C} EUC$ where C is fully controlled by the adversary.

The system must enforce confidentiality, integrity, and authentication even when intermediary servers are malicious.

3.2 *Threat Model and Formal Security Guarantees*

This paper adopts the Dolev–Yao adversarial model, where the attacker A has full control over the communication network and all intermediary nodes N. The adversary may eavesdrop, modify, replay, or forge packets, but cannot break standard cryptographic primitives. A 128-bit security level is assumed, hence: $\varepsilon = 2^{-128}$ is the maximum admissible adversarial success probability.

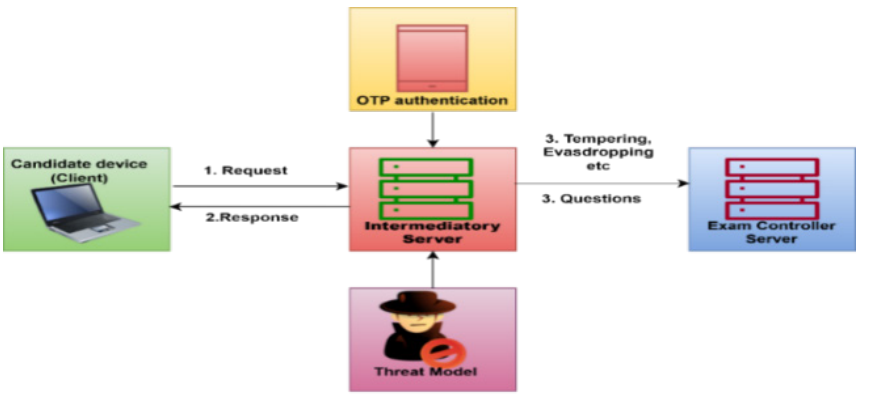

Figure 1
The Architectural Overview of the proposed model

A. *Eavesdropping (Passive Attack)*

Attacker observes ciphertext as A: observe {CQ, CR}. But recovering plaintext violates ChaCha20 IND-CPA security, as given that the $\Pr[A(C) = Q] \leq \varepsilon_{\text{enc}} \leq 2^{-128}$. Thus, confidentiality is preserved.

B. *Packet Modification or Injection (Active Attack)*

Attacker alters ciphertext: $C' = C \oplus \Delta$ and the receiver computes the integrity check: Accept if $SHA256(C') = h$. Forging a valid C'Requires breaking SHA-256 second-preimage resistance as given by the probability $\Pr[SHA256(C') = h] \leq \varepsilon_{\text{int}} \leq 2^{-128}$. Thus, tampering is detected with overwhelming probability.

C. *Replay Attack*

Attacker replays a previously valid pair: $(C, h) \overset{\text{replayed}}{\rightarrow} E$.(C,h) replayed. But TOTP-bound session tokens + ChaCha20 nonces enforce freshness: $\Pr[\text{Replay accepted}] \leq \varepsilon_{\text{replay}} \leq 2^{-128}$.

D. *Impersonation Attack*

Attacker tries to log in: $A \rightarrow E$:Credentials. But authentication uses as $\text{OTP}_U = f_{\text{TOTP}}(K_{\text{shared}}, t)$, so, forging a correct OTP without the secret has a probability: $\Pr[\text{forge OTP}] \leq \varepsilon_{\text{auth}} \leq 2^{-128}$..Thus, impersonation fails.

E. *Man-in-the-Middle (MITM) Attack*

To succeed, an attacker must:

1. Break confidentiality (ChaCha20)
2. Break integrity (SHA-256)
3. Break authentication (TOTP)

Thus: $\varepsilon_{\text{MITM}} \leq \varepsilon_{\text{enc}} + \varepsilon_{\text{int}} + \varepsilon_{\text{auth}} \leq 3 \cdot 2^{-128} \approx 2^{-128}$. MITM attacks are therefore negligible.

Theorem 3.1: (End-to-End Security Guarantee). *In the proposed PSO–ChaCha20–SHA-256–OTP architecture, the probability that any polynomial-time Dolev–Yao adversary A compromises confidentiality, integrity, authentication, or freshness is bounded by the probability* $Pr[A \text{ succeeds}] \leq \varepsilon = 2^{-128}$.

Proof Sketch:

1. **Confidentiality**: ChaCha20 is IND-CPA secure:
   $\Pr[A(C) = Q] \leq 2^{-128}$.
2. **Integrity**: Forging a valid hash requires breaking SHA-256:
   $\Pr[\text{SHA256}(C') = h] \leq 2^{-128}$.
3. **Authentication**: TOTP requires the secret key:
   $\Pr[\text{forge OTP}] \leq 2^{-128}$..
4. **Freshness**: Replays fail due to nonce/counter checks:
   $\Pr[\text{Replay accepted}] \leq 2^{-128}$.

5. **MITM**: Combining all components:
   $\Pr[\text{MITM}] \leq 3 \cdot 2^{-128} \approx 2^{-128}$..

Hence, the theorem holds.

4. Proposed Architecture

This article proposes a secure online examination architecture using an integrated system comprising Particle Swarm Optimization (PSO)- Optimised ChaCha20 encryption, SHA-256-based data integrity verification, and One-Time Password (OTP) authentication to provide confidentiality, integrity, and secure access. The PSO determines the optimal parameters for ChaCha20 encryption by evaluating different particle configurations based on the entropy generated, particle correlations, and encryption time. Once the optimal ChaCha20 parameters are determined, they will be used to encrypt the examination data and protect it during transmission. A SHA-256 Hash will be created from the ciphertext, which will then be used to determine whether the ciphertext has been tampered with before decryption, thereby ensuring its integrity. The OTP authentication will be used to verify the candidate's identity before commencing a secure examination session. Additionally, only encrypted data will be allowed to pass through the servers acting as intermediaries between the candidate and the examination data. The algorithms used for PSO optimisation, and the algorithms used for encryption are outlined in Algorithms 1 and 2, respectively.

Algorithm 1. PSO Optimization
Input: Swarm size N, max iterations $T_{\max}$, inertia w, learning factors c_1, c_2, fitness function $F(\cdot)$
Output: Optimized key configuration $P_{\text{opt}} = (K_{\text{opt}}, N_{\text{opt}}, c_{\text{opt}})$

1. **Initialize swarm**: for each particle $i = 1, \ldots, N$,
2. $x_i(0) = P_i(0) = \{K_i(0), N_i(0), c_i(0)\}, v_i(0) \sim U(-V_{\max}, V_{\max})$.
   Initial fitness: $F_i(0) = F(P_i(0)), pbest_i = x_i(0)$.
   Global best: $gbest = \underset{i}{\text{argmax}} F_i(0)$.
3. **For** $t = 0,1, \ldots, T_{\max} - 1$:
   For each particle i:
   - Velocity update
   - $v_i(t+1) = wv_i(t) + c_1 r_1(pbest_i - x_i(t)) + c_2 r_2(gbest - x_i(t))$,
   - Position update
     $x_i(t+1) = x_i(t) + v_i(t+1)$,
     interpret $x_i(t+1)$ as new config
     $P_i(t+1) = \{K_i(t+1), N_i(t+1), c_i(t+1)\}$,
   - Evaluate fitness
     $F_i(t+1) = F(P_i(t+1))$,
   - Update personal best
   - if $F_i(t+1) > F(pbest_i)$ then $pbest_i \leftarrow x_i(t+1)$, update global best
6. if $F_i(t+1) > F(gbest)$ then $gbest \leftarrow x_i(t+1)$.
7. **Output optimized configuration**:
   $P_{\text{opt}} = gbest = (K_{\text{opt}}, N_{\text{opt}}, c_{\text{opt}})$

Algorithm 2: Encryption
Input: Plaintext P, optimized configuration $P_{\text{opt}} = (K_{\text{opt}}, N_{\text{opt}}, c_{\text{opt}})$
Output: Ciphertext–tag pair (C, h).

1. **Block splitting**:
   Split P into n blocks:
   $P = P_0 \parallel P_1 \parallel \cdots \parallel P_{n-1}$.
2. **Keystream generation** (linked to PSO):
   For each $i = 0, \ldots, n-1$,
   $KS_i = \text{ChaCha20Block}(K_{\text{opt}}, N_{\text{opt}}, c_{\text{opt}} + i)$.
3. **Encryption**:
   For each block,
   $C_i = P_i \oplus KS_i$.
   Concatenate:
   $C = C_0 \parallel C_1 \parallel \cdots \parallel C_{n-1}$.
4. **Integrity tag**:
   $h = \text{SHA256}(C)$
5. **Output**:
   $(C, h) = \text{Enc}(P; K_{\text{opt}}, N_{\text{opt}}, c_{\text{opt}})$

5. Experimental Setup and Performance Evaluation

The experimental evaluation of model is performed in a controlled environment, including the same machine and computational environment for all tests at Intel Core i7 with 16 GB of RAM and a 256 GB SSD running on Python. The different payload sizes ranging from 2^{10}, 2^{12}, 2^{15}, and 2^{20} bytes were tested, and NumPy, Pandas, and Matplotlib were used in the implementation for numerical operations, handling of experimental results, and visualization respectively. To compare the cryptographic performance of the proposed architecture, two algorithms were used as representative of widely employed algorithms: (a) AES-256: a commonly used symmetric block cipher for providing confidentiality of communication; and (b) RSA-1024: a commonly used asymmetric algorithm for key exchange.

5.1 *Encryption and Decryption Time*

Table 1 shows that the proposed model is faster than the alternative algorithms currently being considered, AES-256 and RSA-1024, for execution time in encrypting and decrypting data of various sizes, such as 2^{20} bytes of data, which were encrypted in 55.83 ms by the proposed model, in contrast to 344.22 ms by AES-256 and 1865.74 ms by RSA. The encryption/decryption times are shown in Figure 2. Based on these results, it can be concluded that the proposed model will work very well in high-volume, real-time internet-based testing systems.

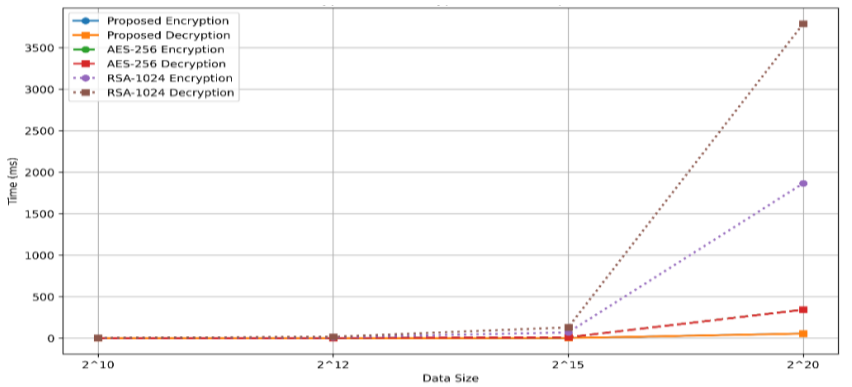

Figure 2
Encryption/Decryption Time (sec)

Table 1
Encryption/Decryption time (ms)

Data Size	Proposed model		AES-256		RSA-1024	
	Encryption	Decryption	Encryption	Decryption	Encryption	Decryption
2^{10}	**0.118**	**0.112**	0.402	0.398	2.481	4.289
2^{12}	**0.402**	**0.39**	1.563	1.55	9.812	17.447
2^{15}	**2.21**	**2.198**	11.37	11.28	68.93	129.67
2^{20}	**55.83**	**54.99**	344.22	342.89	1865.74	3789.31

5.2 *Throughput Analysis*

Throughput measures the amount of data that can be processed in a given time frame and can be mathematically defined using the formula shown in Equation 1.

$$throughput = \frac{Data\ size}{Encryption\ or\ Decryption\ \ time} \quad (1)$$

Using execution time data from Table 1, throughput was calculated. Table 2 demonstrates that the proposed method significantly increases throughput compared to AES-256 and RSA-1024, thereby reducing the time required to transmit examination data and facilitating scalable, real-time encrypted communication among users of online exams.

Table 2
Encryption/decryption throughput (Kb/sec).

Data Size	Proposed model		AES-256		RSA-1024	
	Encr_TP	Decr_TP	Encr_TP	Decr_TP	Encr_TP	Decr_TP
2^{10}	8679.66	9142.85	2547.76	2573.86	412.48	238.69
2^{12}	10186.56	10503.07	2621.3	2643.87	417.63	234.69
2^{15}	14829.86	14909.92	2880.63	2903.54	475.37	252.56
2^{20}	18788.02	19076.69	3047.32	3059.06	562.33	276.82

6. Conclusion and Future Directions

This paper presented a system that utilized PSO-optimized ChaCha20 encryption, SHA-256 for integrity verification, and OTP-based authentication. Experimental testing results showed that the system outperformed AES-256 and RSA-1024, achieving low latency, high throughput, and strong avalanche characteristics. This architecture provides strong assurance of confidentiality, integrity, and authenticity as well as resistance to differential, statistical, and man-in-the-middle attacks. The proposed model is theoretically scalable and computationally efficient; therefore, it may be suitable for future use for securing modern online examination systems. Future studies may consider adding a Trusted Platform Module (TPM) or a Hardware.

References

[1] N. Bisht, B. Pandey, and S. K. Budhani, "Comparative performance analysis of AES encryption algorithm for various LVCMOS on different FPGAs," *World J. Eng.*, vol. 20, no. 4, pp. 669–680 (2023), doi: 10.1108/WJE-05-2021-0281.

[2] D. Goyal, F. Sheth, P. Mathur, and A. K. Gupta, "Discrete mathematical models for enhancing cybersecurity: A mathematical and statistical analysis of machine learning approaches in phishing attack detection," *J. Discret. Math. Sci. Cryptogr.,* vol. 27, no. 2, pp. 569–599 (2024), doi: 10.47974/JDMSC-1893.

[3] S. Dhote, P. Maidamwar, and S. Thakur, "Integrating Blockchain and Multi-Factor Authentication for Enhanced Cloud Security in Certificate Verification Systems," In *Proc. 2024 Int. Conf. Distrib. Comput. Optim. Technol. (ICDCOT)*, pp. 1–7 (2024), doi: 10.1109/ICDCOT61034.2024.10516190.

[4] Thomas Langenfeld, "Internet-Based Proctored Assessment: Security and Fairness Issues," *Educ. Meas. Issues Pract.*, pp. 1–10 (2020).

[5] B. H. Mutongoza and B. E. Olawale, "Safeguarding academic integrity in the face of emergency remote teaching and learning in developing countries," *Perspect. Educ.*, vol. 40, no. 1, pp. 234–249 (2022), doi: 10.18820/2519593X/PIE.V40.I1.14.

[6] A. Noonia, D. Thakral, P. Mathur, F. Sheth, H. Shaikh, and A. K. Gupta, "A discrete mathematical model and cryptography for secure medical image analysis: Encrypted chest X-ray classification," *J. Discret. Math. Sci. Cryptogr.*, vol. 28, no. 5, pp. 1473–1486 (2025), doi: 10.47974/JDMSC-2146.

[7] M. U. Bokhari, S. Afzal, A. Varshney, I. Khan, and Z. Khan, "Enhanced ChaCha20 for next-generation cryptography : A secure and efficient lightweight encryption scheme," *J. Inf. Optim. Sci.,* vol. 47, no. 2, pp. 753–776 (2026), doi: 10.47974/JIOS-2114.

[8] R. K. Muhammed, R. R. Aziz, A. A. Hassan, A. M. Aladdin, S. J. Saydah, T. A. Rashid, and B. A. Hassan, "Comparative Analysis of AES, Blowfish, Twofish, Salsa20, and ChaCha20 for Image Encryption," *Kurdistan J. Appl. Res.,* vol. 9, no. 1, pp. 52–65 (2024), doi: 10.24017/science.2024.1.5.

Received January, 2026