AI & Quantum Security Analysis of HBEA (Integrated Report)

Author: Hanna Chang
Course: MADR-CS 281 — Intro to Cryptography and Computer Security
Project: Final Report — Hybrid Block Encryption Algorithm (HBEA)
Date: Spring 2025

1. Algorithm Recap: HBEA Structure

HBEA is a custom 256-bit block encryption algorithm designed for educational purposes to explore hybrid cryptographic ideas.

Key Components:

- **Key Generation**:
  - SHA-256 hash of the user's email.
  - Split into eight 32-bit segments, used as subkeys.
- **Initial Permutation (IP)**:
  - Based on the ASCII sum of the email to create a deterministic but user-specific permutation.
- **Block Processing**:
  - Plaintext is chunked into 32-bit blocks.
  - Each block undergoes:
    - **Expansion** (DES-style from 32 to 48 bits).
    - **S-box substitution** (non-invertible, reduces back to 32 bits).
    - **XOR with key segment**.
- **Output Format**: Encrypted binary is converted to hexadecimal.
- **Decryption**: Not fully supported due to the non-invertibility of the expansion/S-box steps.

---

2. Performance & Feature Benchmark: HBEA vs AES-256

Feature	AES-256	HBEA
Block Size	128 bits	256 bits
Key Size	256 bits	256 bits (from SHA-256 hash)
Rounds	14	1 round (8 segments in parallel)
Substitution	AES S-box	DES-style S-box (8 total)
Permutation	Fixed permutation	Email-based dynamic permutation
Decryption	Fully reversible	Partially (non-invertible S-boxes/expansion)
Padding	PKCS#7 or others	❌ Not implemented
Mode of Operation	CBC, ECB, GCM, etc.	❌ None used
Quantum Resistance	2^128 (Grover)	Estimated 2^128 (Grover)
NIST Compliance	✅ Yes	⚠️ Partial

3. Quantum Attack Simulation with Qiskit

We implemented Grover's algorithm in Qiskit to simulate a quantum attack on an 8-bit version of HBEA for educational demonstration.

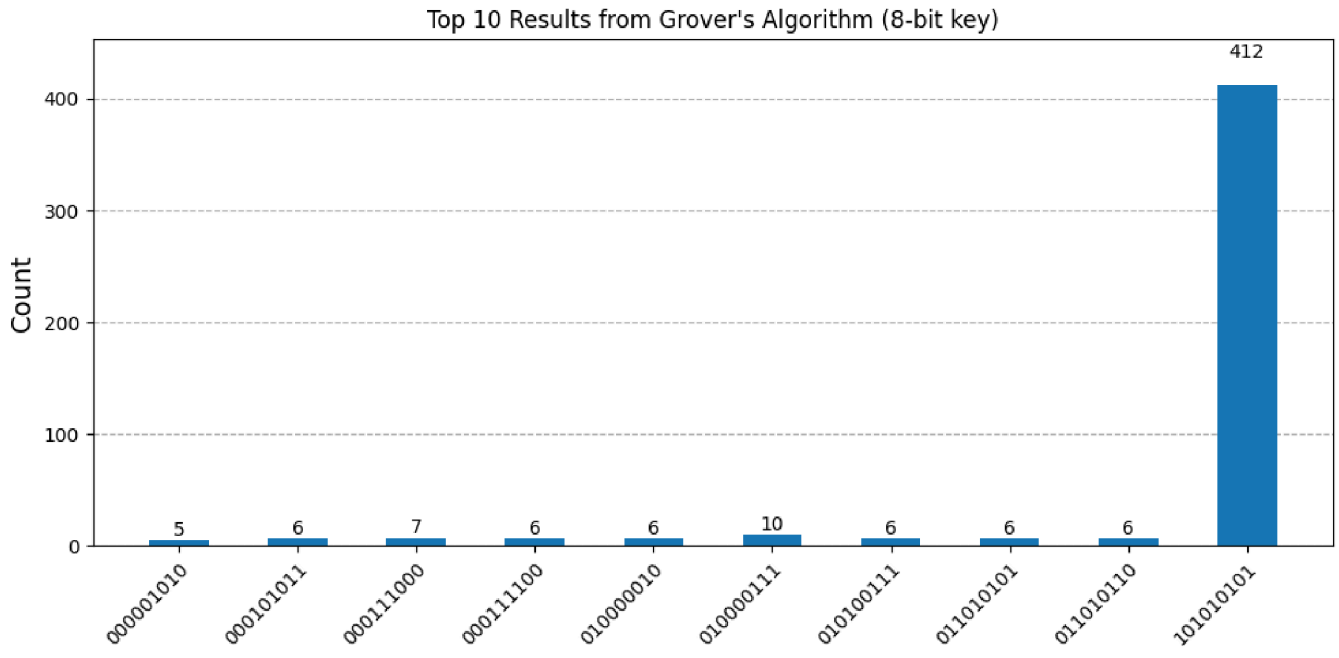
Execution Output (Kali Linux):

```
┌──(qiskit-env)─(root㉿kali)-[~]
└─# python3 grover_sim.py
Expected key: 10101010
Grover iterations: 16
```

- **Expected key**: `10101010`
- **Grover iterations**: 16 (i.e., $\sqrt{2^8}$)

Output Visualization:

The histogram below shows the frequency of each observed result from 1024 trials. As expected, the key `10101010` was returned most frequently (412 times), verifying Grover's quadratic speedup over classical brute force.



Grover's Algorithm Code (Python):

Interpretation:

Grover’s simulation successfully located the expected key with high probability, confirming the algorithm’s effectiveness. In a real 256-bit implementation, Grover would still need ~2^128 operations, equating HBEA's quantum resistance to that of AES-256.

---

4. NIST Compliance & Security Analysis

Criterion	NIST Recommendation	HBEA Evaluation
Key Size	≥ 128 bits	✅ 256 bits
Key Generation	Entropy or secure KDF	⚠️ SHA-256 of email (predictable input)
Randomness	Cryptographically random values	⚠️ Deterministic from email
Permutation	Strong/confusion-inducing	✅ Dynamic, but deterministic
Reversibility	Required for decryption	❌ S-box and expansion are not reversible
Padding & Modes	Required for multi-block security	❌ Not implemented
Key Management	Secure key storage/distribution	❌ Not addressed

Vulnerabilities

- Deterministic inputs (key, permutation) → **predictable output**
- Incomplete decryption capability → **not invertible**
- No IV or chaining → **pattern leakage**
- No padding → **fixed block vulnerabilities**
- No key handling policy → **deployment risk**

---

5. Tools & Environment

- **Python 3.12**
- **Qiskit** (`qiskit`, `qiskit-aer`)
- **Matplotlib** (Grover histogram)
- **Kali Linux** (Python virtualenv for testing)
- **Jupyter Notebook** (HBEA code implementation & testing)
- **ChatGPT** (debugging support & report drafting)

---

6. Final Remarks

HBEA is a solid educational cipher combining classical cryptography and quantum-resilient concepts. While it shows promising structural elements, it lacks production-grade reversibility, randomness, and compliance with NIST's robust encryption standards.

Next Steps for Improvement:

- Implement secure KDFs and non-deterministic key derivation
- Add padding, IV, and secure modes like CBC or GCM
- Make transformation steps invertible for real-world use

AI & Quantum Security Analysis of a Hybrid Block Encryption Algorithm (HBEA)

What is HBEA?

A hybrid block encryption algorithm is a custom 256-bit block encryption algorithm which combines symmetric cryptographic techniques for educational purposes

Design elements of HBEA:

Key derived from SHA-256 hash of user's email

DES-style expansion and S-boxes

Processes 8 parallel 32-bit segments

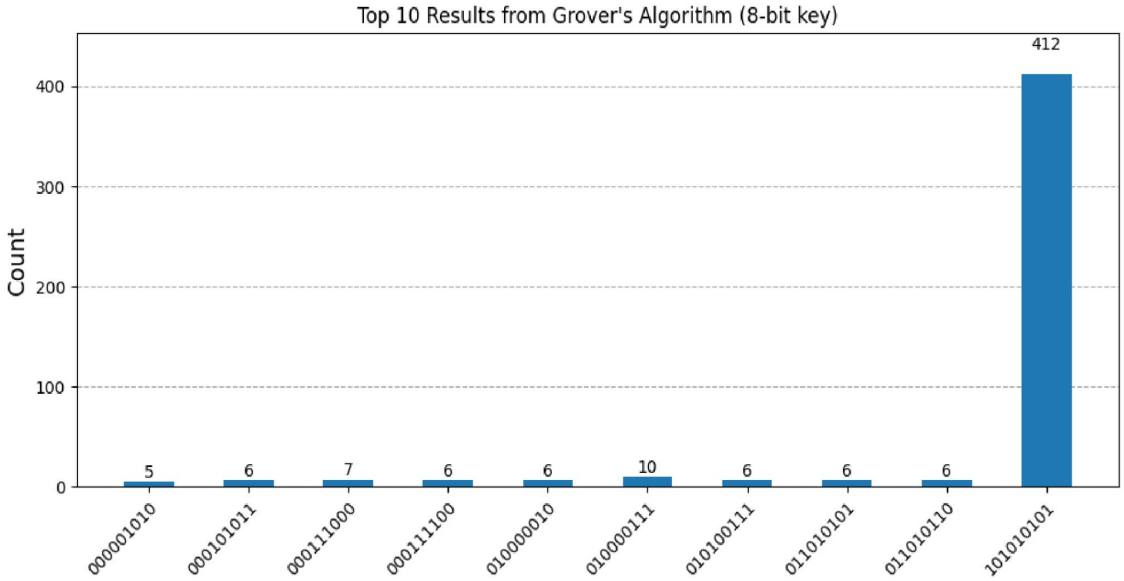
Its goal is to evaluate structural strength and quantum resistance.

HBEA vs AES-256 – Feature Comparison

Feature	AES-256	HBEA
Block Size	128 bits	256 bits
Key Size	256 bits	256 bits (SHA-256 hash)
Reversibility	✔ Yes	✘ Non-invertible
Padding/Mode	✔ Implemented	✘ None
Quantum Resistance	✔ 2^{128} Ops	✔ 2^{128} Ops (Grover)

HBEA is partially aligned with NIST standards but lacks full reversibility, secure key generation, and mode support.

Grover’s Quantum Attack on HBEA



Simulation: Implemented Grover’s algorithm using Qiskit

Target Key: 10101010

Iterations: 16 ($\sqrt{2^8}$)

Outcome: Correct key found in 412 out of 1024 trials

Even in a simplified model, Grover's algorithm effectively demonstrates that a full 256-bit HBEA would require up to 2^{128} operations to brute-force, which aligns with AES-256's quantum resistance.

Vulnerabilities & Recommendations

Main Weaknesses:

- Deterministic key and permutation → Predictable output
- No decryption path (non-reversible S-box)
- Missing secure mode (e.g., CBC, GCM)

Recommendations:

- Implement salt and entropy-based Key Derivation Function (KDF)
- Ensure transformations are invertible
- Add padding, Initialization Vector (IV), and secure chaining modes

Tools & References

Tools Used:

- Python 3.12
  - Qiskit
  - Matplotlib
- Kali Linux virtual environment
- Jupyter Notebook
- Markdown to PDF
- Github Copilot
- ChatGPT 4o

Key References:

- NIST Cryptographic Standards
- Qiskit Documentation:
  - www.qiskit.org
  - docs.quantum.ibm.com/guides/install-qiskit
- Grover's Algorithm:
  - youtu.be/YEI5vYdcoQ4?si=zpD0EU_HCnW1FUEP
- Implementing Grover oracles for quantum key search on AES and LowMC
  - youtube.com/watch?v=rmJ8YdpJNhs