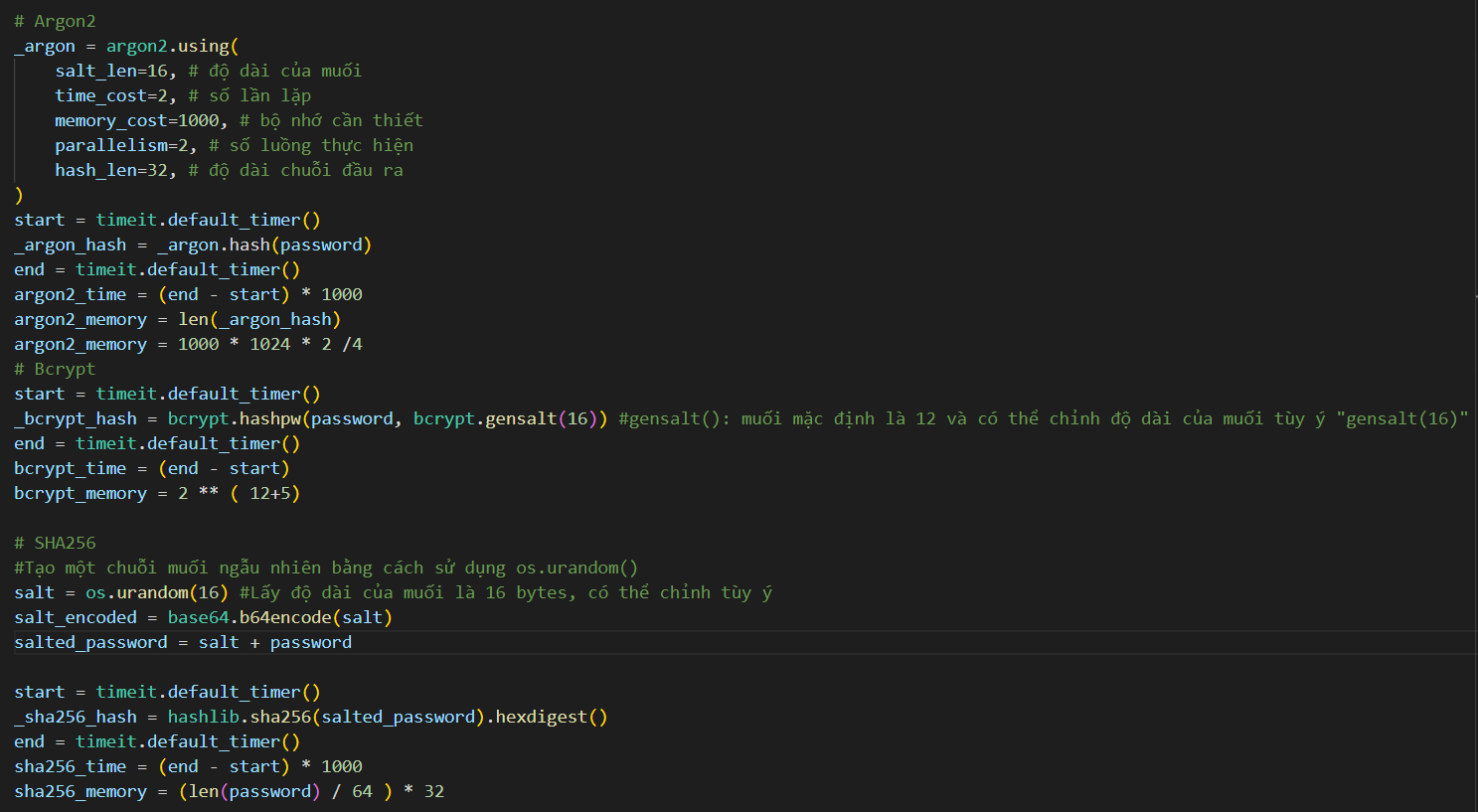
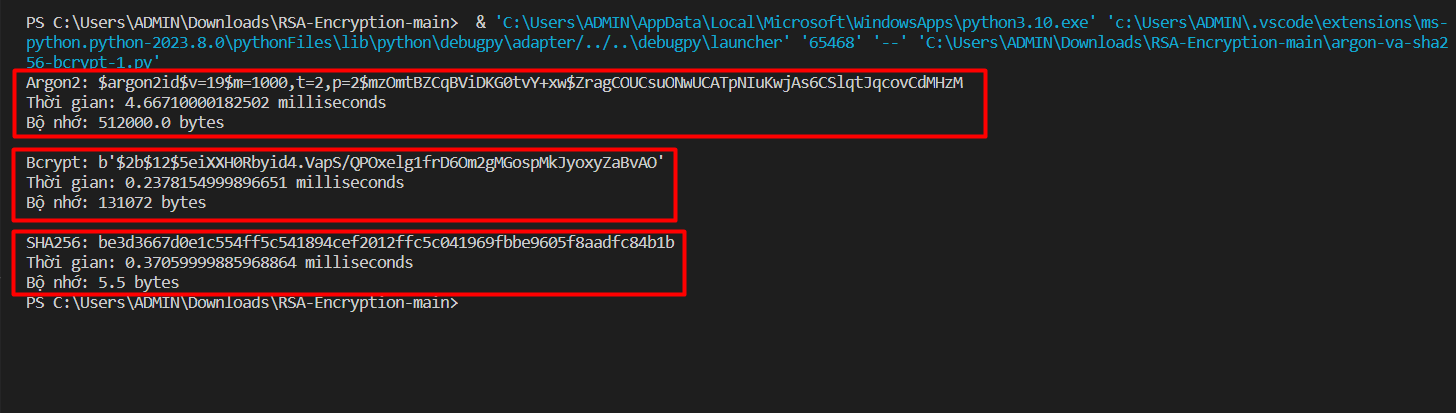
|  |  |  |  |
| --- | --- | --- | --- |
|  | Argon2 | Bcrypt | SHA-256 |
| Thuật toán | Argon2 là một thuật toán băm mật khẩu dựa trên Blake2 và được thiết kế để làm chậm quá trình băm và tăng độ bảo mật bằng cách sử dụng một lượng lớn bộ nhớ. | Bcrypt sử dụng hàm băm Blowfish và được thiết kế để làm chậm quá trình băm mật khẩu và tăng độ bảo mật bằng cách sử dụng một lượng lớn bộ nhớ. | SHA-256 là một thuật toán băm không có trạng thái, dựa trên hàm băm đệ quy trên cơ sở của lý thuyết thông tin. |
| Tính bảo mật | Argon2 được thiết kế để chống lại các cuộc tấn công bằng từ điển, tấn công bằng lực brute và tấn công đa nhân. | Bcrypt được thiết kế để chống lại các cuộc tấn công bằng lực brute và các tấn công từ điển. | SHA-256 là một thuật toán băm mạnh và được sử dụng rộng rãi trong các ứng dụng bảo mật, nhưng nó dễ dàng bị tấn công bằng lực brute và tấn công từ điển so với Argon2 và Bcrypt. |
| Tốc độ băm | Argon2 được thiết kế để làm chậm quá trình băm bằng cách sử dụng một lượng lớn bộ nhớ. Chậm hơn so với SHA-256 và Bcrypt. | Bcrypt cũng được thiết kế để làm chậm quá trình băm bằng cách sử dụng phương pháp "phân tán" và một tham số gọi là "factors". Tuy nhiên, tốc độ băm của Bcrypt nhanh hơn so với Argon2 và chậm hơn so với SHA-256. | SHA-256 là một thuật toán băm nhanh và có thể băm một lượng lớn dữ liệu một cách nhanh chóng. Tốc độ băm của SHA-256 nhanh hơn so với Argon2 và Bcrypt. |
| Sử dụng bộ nhớ | Argon2 được thiết kế để sử dụng một lượng lớn bộ nhớ, để làm chậm quá trình băm và tăng độ bảo mật. Sử dụng kỹ thuật "phân tán bộ nhớ". Nên dùng nhiều bộ nhớ hơn  SHA-256 và Bcrypt. | Bcrypt cũng sử dụng một vài lượng bộ nhớ để lưu trữ, tuy nhiên lượng bộ nhớ được sử dụng ít hơn so với Argon2 và nhiều hơn SHA-256. | SHA-256 không đòi hỏi một lượng lớn bộ nhớ để hoạt động. Sử dụng ít bộ nhớ hơn Argon2 và Bcrypt |
| Hiệu suất | Argon2 có thể chậm hơn SHA-256 trong việc tính toán hash mật khẩu, do sử dụng các kỹ thuật bảo mật phức tạp hơn. Tuy nhiên, việc sử dụng các tham số hiệu suất của Argon2 cho phép điều chỉnh hiệu suất và độ an toàn của thuật toán. | Trong trường hợp thời gian hash dưới 1s, bcrypt có thể hoạt động nhanh hơn Argon2. Nếu tính an toàn là ưu tiên hàng đầu, Argon2 là lựa chọn tốt hơn. Tuy nhiên, nếu hiệu suất là ưu tiên hàng đầu và thời gian hash dưới 1s, Bcrypt có thể là lựa chọn tốt hơn. | SHA-256 có hiệu suất cao hơn so với Argon2 và Bcrypt, nhưng không có khả năng điều chỉnh độ an toàn và hiệu suất của thuật toán. |
| Kích thước đầu ra | Argon2 cho phép tạo ra các chuỗi băm (hash) với độ dài tùy ý. | Bcrypt luôn tạo ra các chuỗi băm có độ dài cố định là 192 bit. | SHA-256 luôn tạo ra các chuỗi băm có độ dài cố định là 256 bit. |
| Độ tin cậy | Argon2 được coi là một thuật toán hash mật khẩu đáng tin cậy hơn so với SHA-256 và Bcrypt. | Bcrypt được coi là một thuật toán hash mật khẩu đáng tin cậy. Tuy nhiên độ tin cậy của Bcrypt không bằng so với Argon2. | SHA-256 vẫn được coi là một thuật toán hash mật khẩu đáng tin cậy. |

-Code so sánh Argon2 với Bcrypt và SHA-256





* Kết luận: Argon2, SHA-256 và Bcrypt đều là các thuật toán băm mật khẩu được sử dụng rộng rãi trong các ứng dụng bảo mật.

-SHA-256 là một thuật toán băm mật khẩu đơn giản và nhanh chóng, phù hợp để sử dụng trong các trường hợp đòi hỏi tính nhanh và sử dụng ít bộ nhớ nhưng không đảm bảo độ bảo mật cao so với Argon2 và Bcrypt.

-Bcrypt là một thuật toán băm mật khẩu, tốc độ băm của Bcrypt nhanh hơn so với Argon2 và chậm hơn so với SHA-256. Bcrypt cũng sử dụng ít bộ nhớ hơn so với Argon2.

-Argon2 là một thuật toán băm mật khẩu được coi là nâng cao hơn so với Bcrypt và SHA-256, vì nó sử dụng các kỹ thuật mới nhất để tăng độ bảo mật và khó khăn hơn cho các cuộc tấn công.

- Việc lựa chọn thuật toán băm mật khẩu phù hợp phụ thuộc vào yêu cầu của ứng dụng và mức độ bảo mật được yêu cầu. Trong các trường hợp đòi hỏi độ bảo mật cao hơn, Argon2 là lựa chọn tốt nhất. Tuy nhiên, trong những trường hợp đòi hỏi tính nhanh và độ bảo mật ở mức trung bình, Bcrypt hoặc SHA-256 có thể là những lựa chọn phù hợp hơn.