**Session Implementation in Different Authentication Libraries**

Lucia:

* Lucia uses a separate ID and secret to prevent any possibility of a timing attack. The secret is hashed before storage to minimize the impact of breaches and leaks.
* The secret hash is stored as a raw binary value, but it can also be stored as a string (by hexadecimal or base64 encoding).
* The IDs and secrets are generated from a random byte array and then encoded into a string. The random byte array is generated from a cryptographically secure random source and is immune to offline brute-force attacks.
* The secret is hashed using SHA256 and is immune to offline brute-force attack.
* To validate a sessions token, parse out the ID and secret, get the session with the ID, check the expiration, and compare the secret against the hash. Use constant-time comparison for checking secrets and derived hashes.

**Note: An expiration time should be set for all sessions**. **If the aim is to keep active users signed in, then an inactivity timeout should be set.**

* The Session should be stored in a **HttpOnly, Secure,** cookie with **SameSite**set to **Lax**. If you want a persistent session, set a new cookie periodically.

**Note: It's important to note that** **using a HttpOnly cookie does not make you immune to targeted XSS attacks. For native applications, use the device's built-in secure storage.**