**OpenSSL vs BoringSSL**

**Initial Remarks**

1. Development and Maintenance: OpenSSL has been around since 1998 and has a larger development community. It is maintained by the OpenSSL Software Foundation and receives contributions from various individuals and organizations. BoringSSL, on the other hand, is a fork of OpenSSL that was created by Google in 2014. It is specifically designed to be used internally by Google products, and its development is primarily driven by Google engineers.
2. Focus on Simplicity and Security: BoringSSL aims to be a minimalistic and simplified version of OpenSSL. It has a smaller codebase and a reduced feature set, focusing on providing only the essential cryptographic functions with a cleaner design. BoringSSL's goal is to improve code maintainability and security by minimizing the attack surface and reducing the risk of bugs and vulnerabilities.
3. API Compatibility: While BoringSSL started as a fork of OpenSSL, it has diverged significantly from the original codebase over time. BoringSSL introduced various API changes and removed certain legacy features and options present in OpenSSL. As a result, applications developed using OpenSSL may require modifications to work with BoringSSL.
4. Integration with Google Projects: BoringSSL has been optimized and tailored for Google's specific use cases and requirements. It has deeper integration with Google's software stack and is used in many Google products, such as Chrome, Android, and others. BoringSSL may include features and optimizations that are specific to Google's needs but not available in OpenSSL.
5. Licensing: OpenSSL historically used the OpenSSL License, which is considered to be permissive but has caused some licensing concerns in the past. BoringSSL, however, uses a more standard and simplified license called the "OpenSSL License," which is based on the Apache License 2.0. This change was made to address some of the licensing ambiguities and compatibility issues associated with the original OpenSSL License.

It's worth noting that while BoringSSL was initially intended for internal use at Google, it is open-source and can be used by others outside of Google. However, due to its more specialized nature, BoringSSL is generally recommended for projects closely tied to Google's ecosystem, while OpenSSL remains a widely adopted choice for general cryptographic needs in various applications and systems.

**No Type 1 Discrepancies (Discrepancies that cause differing return codes) detected with either BoringSSL or OpenSSL as the target in either of these directories:**

1. 20220421\_165232\_openssl-1.0.2h-boringssl-f0451ca-update-1\_tgt-0\_all
2. 20220421\_213203\_openssl-1.0.2h-boringssl-f0451ca-update-1\_tgt-1\_all
3. 20220511\_171507\_openssl-1.0.2h-boringssl-f0451ca-update-1-rank-1-82802\_tgt-0\_all
4. 20220511\_212025\_openssl-1.0.2h-boringssl-f0451ca-update-1-rank-1-82802\_tgt-1\_all
5. 20220511\_224405\_openssl-1.0.2h-boringssl-f0451ca-update-1-rank-2-2c0a0\_tgt-0\_all
6. 20220512\_024432\_openssl-1.0.2h-boringssl-f0451ca-update-1-rank-2-2c0a0\_tgt-1\_all