Towards a central, distributed and secure default cryptography parameter set

Kai Mindermann
University of Stuttgart

Stefan Wagner
University of Stuttgart

30th Crypto Day, 28/29 March 2019

Cryptography encryption algorithms like the Advanced Encryption Standard (AES) have to be configured with a specific key length. Usually it is required to specify a block mode (e.g. Galois/Counter Mode (GCM)) and a padding algorithm. All these choices influence the security fundamentally. In practice, it has been shown that developers rely on outdated documentation and tutorials to set these parameters [4, 2]. Therefore, the approach to let developers choose parameters often leads to insecure applications.

A better approach would be to include a secure default parameter set with all choices made in every cryptography library.

Furthermore, updating these parameters is not always as simple as for encryption of a transport channel, for instance with Transport Layer Security (TLS). With TLS several algorithms can be used, of which one is determined between the client and the server for the duration of one connection, and prioritized, again both by the client and the server. In contrast, for example encrypting files currently requires that the parameters stay the same, otherwise older data could not be decrypted or other not updated applications could not decrypt the data. This can be countered by implementing separate backwards compatible layers in the software, but that increases the maintenance costs and reduces the reliability of the software.

A better approach would be to store the used parameters alongside the encrypted data and retrieve this configuration data for decryption from the data rather than from the independent application source code or environment.

1 Secure Crypto Configuration (SCC)

The general idea of the SCC can be described with the following process (see also fig. 1):

1. Each year the Internet Engineering Task Force (IETF) decides in cooperation with other institutions like the Bundesamt für Sicherheit in der Informationstechnik (BSI) or the National Institute of Standards and Technology (NIST) what cryptography parameters can be considered secure for which kind of classified information (TOP SECRET to UNCLASSIFIED). The consortium also specifies for each parameter set how long it is considered secure (from the time of publication).

crypto day matters 30 (2019)
Beiträge vom 30. Kryptotag, 28.-29.03.2019, Mozilla, Berlin.
doi:10.18420/cdm-2019-30-25

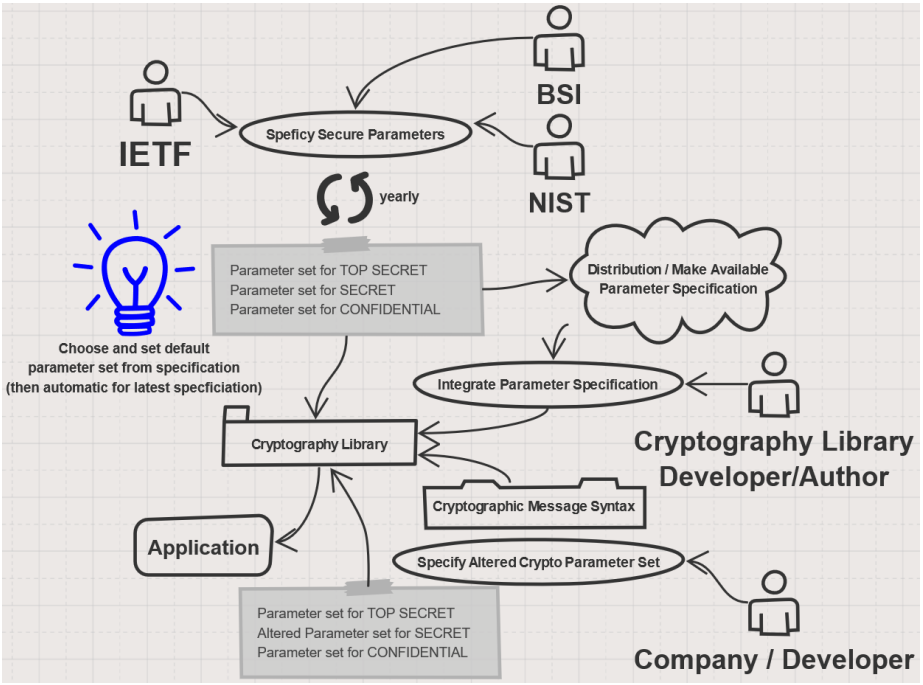

Figure 1: Sketch for the process of the Secure Crypto Configuration (SCC)

2. The consortium publishes the SCC in a machine and human readable format including a version (YYYY-MM).

3. The machine-readable format is then used to update the default parameter selection of cryptography libraries.

4. Additionally, companies or single developers can derive their own SCC and override the default configuration for their applications.

This ensures that available cryptography implementations always use a configuration that is currently considered secure. It also prevents applications from using insecure defaults and to use hard coded (or self managed) parameter sets.

2 Cryptographic Message Syntax (CMS)

One caveat with the updateable parameter set is that the default implementation changes over time, regarding the output of its cryptography algorithms. To prevent applications from being incompatible with newer SCC, the cryptography libraries must integrate the parameter set in the encrypted data so it can be read from the encrypted data instead of from some independent location that has to be kept synchronized with possible parameter sets of the old data.

There already exists a standard for this, the Cryptographic Message Syntax (CMS) [3]. It defines data structures for several kinds of cryptography algorithm outputs (like signed or encrypted data) and includes special fields for the used parameters. With the default usage of such a standard, future changes

to the parameter sets would not require additional software layers that handle previously used cryptography parameter sets, but instead they are read from the data and used to configure the cryptography library on the fly.

3 Conclusion and Future Work

The proposed SCC would create a common security level. It would also prevent outdated documentation and tutorials, because they could refer to a `LATEST-Configuration` in the code, which would always use the latest cryptography parameter set available for the used cryptography library. Additionally, it would make security audits a lot easier, because only the configuration would have to be checked and not hard coded parameters and the custom build backwards compatibility layers. The new approach does also not prevent power users from changing the configuration themselves, because it can be changed on the application side which overrides the default SCC.

Institutions like the BSI have to work together with the IETF to ensure their current cryptography recommendations [1] are integrated into the SCC. Yet, they could also offer their own *official* parameter set in the same format. Not only these institutions need to work together, but also authors of cryptography libraries need to integrate mechanisms to utilize the SCC and CMS.

Performance is important and depends on the algorithm and its parameter choices, but in our opinion it is more important to increase the security first, hence the focus on default behavior, and then optimize for performance if necessary. There are areas where this approach does not improve the security. For example the management of keys (creating, storing, distributing, removing) is another important aspect of cryptography, but it is out of scope for this approach. Still, we believe that the proposed Secure Crypto Configuration (SCC) improves the status-quo of cryptography.

References

[1] BSI. *Kryptographische Verfahren: Empfehlungen und Schlüssellängen. Technische Richtlinie*. Tech. rep. TR-02102-1. Bundesamt für Sicherheit in der Informationssicherheit, June 2018. URL: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf?__blob=publicationFile&v=8.

[2] F. Fischer et al. "Stack Overflow Considered Harmful? The Impact of Copy Paste on Android Application Security". In: *2017 IEEE Symposium on Security and Privacy (SP)*. May 2017, pp. 121–136. DOI: 10.1109/SP.2017.31.

[3] R. Housley. *Cryptographic Message Syntax (CMS)*. INTERNET STANDARD RFC5652. Internet Engineering Task Force, Sept. 2009. DOI: 10.17487/RFC5652. URL: https://tools.ietf.org/html/rfc5652.

[4] Kai Mindermann and Stefan Wagner. "Usability and Security Effects of Code Examples on Crypto APIs". In: *2018 16th Annual Conference on Privacy, Security and Trust (PST)*. Belfast, Northern Ireland, United Kingdom: IEEE, Aug. 28, 2018, pp. 1–2. ISBN: 978-1-5386-7493-2. DOI: 10.1109/PST.2018.8514203.