SANDWICH

Sandwich Open Source Library

Easy Access to Agile Cryptography for Your Applications

UPDATING CRYPTOGRAPHY WITHOUT BUSINESS DISRUPTION

Modern cryptography management is becoming an integral part of risk and compliance processes as regulators mandate the adoption of cryptographic agility and migration to Post-Quantum Cryptography. A crypto-agile architecture enables applications to quickly adapt to new cryptographic primitives and algorithms. It ensures seamless integration with existing IT infrastructure without disrupting or slowing down business operations.

However, implementing this adaptability within applications leads to serious challenges for developers and cryptographic experts:

- **Crypto-agility is complex:** Requiring every business to create unique agile cryptography solutions is impractical and inefficient. Developers, cryptography experts, and security teams already have full workloads.
- **Crypto-agility increases security:** Quantum threats still lie ahead, but vulnerable cryptography must be replaced today. Despite having known issues MD5, SHA1, and 3DES are still widely deployed due to a lack of crypto-agility.
- **Crypto-agility requires deep visibility:** Achieving that visibility is not easy. To reduce business continuity risks it is important to understand cryptographic dependencies.
- **Crypto-agility requires correct configuration:** Next generation cryptography can still be misconfigured. Crypto-agility must include a way to analyze newly updated cryptography for configuration errors, performance issues, and vulnerabilities.
- **Crypto-agility should not constrain developers:** Without support for modern programming languages like C, C++, Python, Go, and Rust, developers may resist adoption. Similarly, any solution should be library agnostic to appeal to developers.

SANDWICH MODERNIZES CRYPTOGRAPHY MANAGEMENT

Sandwich is an open source, unified API that removes the complexity of cryptographic libraries for developers and instead lets them leverage the API to do the heavy lifting. Sandwich supports languages, such as C, Rust, Python, Go, and cryptographic libraries OpenSSL and BoringSSL, as well as libOQS, to easily enable post-quantum cryptography.

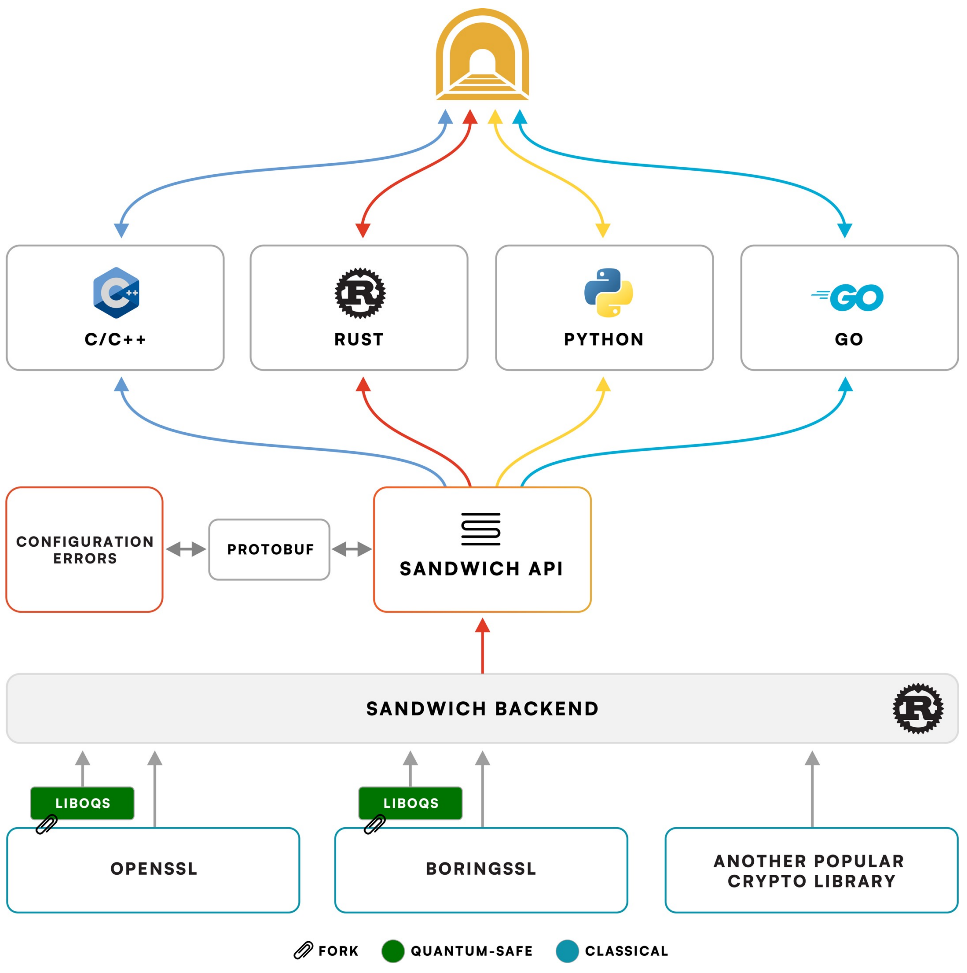
By supporting multiple languages and popular cryptography backend providers, the Sandwich API significantly reduces the amount of work developers have to do, allowing them to use cryptography securely in any language and enabling access to best practice cryptography without having to learn the use of specific libraries.

When these libraries change, developers can simply update Sandwich and it will integrate the new changes on the back end, allowing continued use of the familiar, streamlined API.

Because Sandwich abstracts cryptography from the application code it creates a crypto-agile architecture that makes it easy for security teams to update and replace algorithms whenever they need to.

SANDWICH BINDINGS

- Simple API
- Misuse prevention
- Minimal overhead



SANDWICH BACKEND

- Link to a cryptography library at compile-time
- Support liboqs on top of OpenSSL and BoringSSL
- Easy to extend for other libraries

HOW TO USE SANDWICH

Sandwich offers a range of industry-standard protocols, making it easy to adopt and integrate tried-and-tested cryptographic approaches. Sandwich adds another popular library to the backend which enables access to Post-Quantum Cryptography (PQC), allowing developers to experiment with cutting-edge cryptographic techniques. Sandwich provides bindings for various programming languages, separating the choice of a cryptographic library from the choice of the application language. This flexibility allows developers to work in their preferred language while harnessing the full capabilities of Sandwich.

Here are the steps to implement Sandwich:

- Decide which protocol you want to use (e.g., TLS 1.3)
- Choose the implementation (e.g., OpenSSL+libOQS)
- Sandwich composes them into a Sandwich object (in this case, a secure tunnel).

The protocols and implementations that are available at runtime are dependent on the choices that were made when compiling the Sandwich library. Developers will be able to compose their own sandwich, for choices and options they would like to have at runtime, giving them the ability to change configurations without having to re-compile their code.

HOW IS SANDWICH DIFFERENT?

One key distinction of Sandwich is that it does not perform cryptography itself. Instead, it allows developers to integrate their own trusted backend, providing complete control and confidence in the underlying cryptographic operations. This flexibility sets Sandwich apart, ensuring that developers can work with their preferred cryptographic implementations while leveraging the benefits of Sandwich's powerful framework.

Sandwich provides a significant advancement in capabilities compared to traditional libraries like OpenSSL and BoringSSL. With Sandwich, developers have the power to choose their desired protocol, such as TLS, and implementation, such as OpenSSL, within a unified object (i.e., a sandwich). This abstraction shields developers from the complexities of these choices, offering a streamlined and hassle-free experience which significantly reduces the implementation complexity.

In contrast, traditional libraries primarily offer predefined functions without the same level of flexibility and customization. Sandwich breaks down barriers and reduces effort, making it significantly easier for developers to leverage cryptographic capabilities and integrate them seamlessly into their applications. Sandwich combines the strengths and functionalities of multiple libraries, empowering developers with a rich selection of cryptographic tools at their disposal.

THE BENEFITS OF USING SANDWICH

- **Modernize Cryptography Management:** Provide your operational teams and developers the capabilities they need to oversee, manage, and update cryptography throughout your organization.
- **Deploy Applications Faster:** Remove cryptography management bottlenecks from your DevOps pipeline and allow your developers to easily self-serve best practice cryptography without waiting for Subject Matter Experts.
- **Ease the Transition to Post-Quantum Cryptography:** Make migration to new Post Quantum Cryptography easier by embedding a crypto-agile architecture into your applications.
- **Reduce Risk of Data Breaches:** Reduce exposure by creating a mechanism to easily update cryptography that has become vulnerable.
- **Reduce Time to Resolution:** Update your cryptography faster by centralizing control.
- **Reduce Compliance Failures for FIPS, FedRAMP, PCI-DSS, and more:** See and resolve any compliance failures in your applications. Update effortlessly, even when compliance requirements evolve.
- **Enable Large Scale Adoption of Crypto-Agility:** Bring everyone along by allowing developers to use their preferred languages and libraries.

Sandwich exemplifies the SandboxAQ approach to crypto-agility, allowing applications to access cryptographic services in a way that allows their algorithms, protocols and backend cryptographic libraries to be changed, without breaking applications.

In the future, SandboxAQ will be releasing more functionality in Sandwich, in particular for key-management. We also welcome feedback, bug reports, feature requests and contributions from the community. Visit our website at https://www.sandboxaq.com/solutions/sandwich and contact us at Sandwich@sandboxaq.com.