Quarkslab

Shield

A SHIELD FOR YOUR SOFTWARE AND A VAULT FOR YOUR SECRETS

Software is everywhere into our daily lives from our coffee machine, tv, car, phones, planes, healthcare devices to nearly any equipment we interact with on a single day.

Organisations depend on software to deliver their services or goods and carry their business and thus all the software dependent revenues are under a permanent and imminent cyber-threat.

As financial incentives are constantly increasing for hackers, it is to the defender to deter the attacker by making the cost of the attack superior to the potential revenues.

In-app protection software such as application shielding, white-box cryptography and digital vault functionalities contributes to significantly increase the cost of an attack to thwart potential attackers.

A UNIQUE TOOL SUITE TO BRING SOFTWARE IN COMPLIANCE WITH BUSINESS RISKS

KEY BENEFITS

- Protect your revenues by preventing theft, fraud, counterfeiting and hacking
- Prevent unauthorised recovery and disclosure of the source code
- Prevent unauthorised analysis of the software internal functioning
- Prevent theft of passwords, secrets and keys
- Prevent theft of Intellectual Property
- Extensive services and support to facilitate securing your business
- Seamless integration with DevOps and Agile software processes

10, Boulevard Haussmann
75009 Paris France

+33 (0)1 58 30 81 51

contact@quarkslab.com

quarkslab.com
quarkslab.com/blog

App Protection

We protect your software so you can safely deliver your business value

Quarks App Protect provides best-in-class software and data shielding through obfuscation and environmental checks to prevent unauthorised parties to attempt gaining insight, tamper with or even recover the source code from the binary.

Best-in-Class Application Shielding

Benefit from the **most versatile and top-tier application shielding solution**. With the ability to **combine more than 30 obfuscation passes**, there is always a way to reach the right balance in between security and performance.

KEY FEATURES

- Any combination of 30+ obfuscation passes for the best security/ performance ratio
- Dynamic Protections: anti-root, anti-jailbreak, anti-debug, anti-hooking
- Integrity checks coupled with anti-tampering technologies
- Call-backs to trigger specific actions on detection of anomalies
- Enable Security Officer to observe and ensure that the application complies with security guidelines

TECHNICAL DETAILS

- Obfuscations available for C, C++, Java, Kotlin, specific language on demand
- RASP available for Linux, Windows, Android, iOS, OSX
- Architectures: Mobile, Embedded, desktop (ARM, ARM64, x86, x64)
- Integrations: xCode Visual Studio, Android Studio

Dynamic Protection with Device Trust Assessment Features

A software might find itself running on unvetted devices and thus requires **the ability to defend itself using app shielding technologies**. Quarks App Protect integrity technology (i.e. RASP) **protects the application from tampering attempts**. Combined with obfuscation passes, it prevents unauthorised analysis of the application behaviour or recovering source code via reverse-engineering techniques.

Security must comply with business needs and regulatory standards

Security is here to protect business revenues and enable safe growth. To achieve these goals, Quarks App Protect has been designed with agility and versatility, enabling to reach the required security levels seamlessly and without being an hindrance. **Security officers can design the most appropriate security configuration** and can count on our team of **Subject Matter Experts to support them in achieving their objectives if needed.**

Keys Protection

The art of hiding secret keys in the plain visible sight of a software binary

White-box cryptography combines **methods of encryption and obfuscation to embed secret keys within application code**. The goal is to combine code and keys in such a way that the two are indistinguishable to an attacker.

Hide cryptographic keys in software

Any visual inspection of an unprotected software binary containing a cryptographic key will enable any unskilled attacker to quickly find the key. **White-box Cryptography** is the art of not only **concealing the keys**, but also ensuring a **safe usage** to prevent skillful attackers trying to extract keys through side-channel attacks.

KEY FEATURES

- Hide your secrets and cryptographic keys
- Prevent recovery of secret keys by analyzing the behavior of the application
- Supports derivation of secret keys from a master key
- Supports customization of keys in binaries to generate unique software out of a generic version
- Individual white-box implementations per customer/device

TECHNICAL DETAILS

- Secure integration of a static key or wrapping and unwrapping of derived keys from a master key
- Available algorithms: AES128, ECDSA (NIST P-256), SHA-1/256/3, AES-CMAC. ECDSA-SHA-256 (NIST P-256), other algos on demand
- Architectures: Mobile, Embedded, desktop (ARM, ARM64, x86, x64)
- C/C++ Library

Mandatory solution for all software with encryption

No organisation would be using cryptography if there wasn't an asset to protect. If it is deemed possible that the attacker can gain access to the device or the software, **White-Box Cryptography is then required** to ensure that cryptographic keys are safe from prying eyes.

Unique white-box implementations

To prevent an attacker from building up insights by doing the analysis of a series of white-boxes, Quarks Keys Protect has a unique **white-box generation technology**, meaning that **implementations are unique and not shared among customers.**
With this design, **not even Quarkslab** has access to your keys stored in your generated white-boxes.

Seamless integration

Available as C++ and C software libraries, and in Java via JNI, Quarks Keys Protect can easily be integrated into your project through simple API calls.

Digital Vault

Uniform security for all sensitive data

As soon as a software, i.e. **a mobile application or IoT object,** connects to a remote server it has to authenticate itself and thus handle credentials and authentication tokens. If these elements lack the necessary protection mechanisms and are recovered by an attacker, they can be used for an attack again the overall solution.

Protecting any and all assets

Quarks Digital Vault allows developers to **securely store any kind of sensitive data** used by an application from authentication credentials, password, authentication tokens, serial numbers, etc.

KEY FEATURES

- Unique anti-theft vault thanks to device binding
- Leverage any hardware-based capabilities to provide best-in-class security
- Obfuscated and protected against reverse engineering
- White-box cryptography security
- Android KeyStore integration
- Based on a solid security paradigm: decrypt only what you need

TECHNICAL DETAILS

- Key-value interface to store encrypted files, docs and data.
- C/C++ Library available for Android and Linux
- Architectures: Mobile, Embedded, desktop (ARM, x86, x64)
- Supporting up to 1024 entries (max size for each entry: 10MB)

Unique to each device

The Digital Vault upon its first run builds a fingerprint of the device and **binds itself to the device** such that even if lifted, extracted, from the peripheral, **it cannot be used.**

Hardware and Software Security Combined

Quarks Digital Vault **automatically detects and leverage any hardware security** such as for instance, cryptographic functions offered by a Trusted Execution Environment (TEE). If not available, **it automatically defaults to using software-based cryptography** thus enabling to abstract security and deliver seamless integration with your application.

Seamless integration

Available as C++ and C software libraries, and thus in Java via JNI, Quarks Digital Vault can easily be integrated into your project through simple API calls.