WhiteboxCRYPTO™

Cryptographic key hiding with tunable security and performance

Cryptographic Key Protection

Access to data, information systems, and digital content is commonly protected by encryption. The single point of failure for any crypto system is the instance in which the key is used. This point is easily identifiable in software routines using signature, pattern, and memory analysis. Typically, key extraction attacks against keys coded as literal data arrays in unprotected applications can be successfully completed in a matter of hours.

Microsemi's WhiteboxCRYPTO™ product combines mathematical algorithms, data, and code obfuscation techniques to transform the key and related crypto operations in complex ways requiring deep knowledge in multiple disciplines to attack. Importantly, **the key is never present in static or runtime memory.** Rather, the key becomes an inert collection of data that is useless without the uniquely generated white box algorithm. WhiteboxCRYPTO™ comes in many variants:

WhiteboxAES™ 128, 192, and 256-bit key protection
WhiteboxRSA™ Encrypt/decrypt/sign/verify all key sizes
WhiteboxECC™ p160/192/224/256/384/521 prime curve
WhiteboxSHA™ SHA1, SHA2 224/256/384/512 + HMAC
Whitebox3DES™ DES and 3DES for all key sizes
WhiteboxOXD™ Obfuscation for sensitive data transfer
WhiteboxSSL™ provides protection from known and future OpenSSL vulnerabilities and attack vectors
WhiteboxFFC™ Finite Field Crypto/Diffie-Hellman-Merkle
WhiteboxJCE™ Facilitates the use of third party implementations through Java Cryptography Extension (JCE)
WhiteboxTLS™ Transport Layer Security (TLS) protocol
WhiteboxCMLA™ CMLA key derivation function (KDF)
WhiteboxNK108™ NIST SP800-108 KDF

Whitebox Key Transformation

To protect encrypted information, it is imperative that the key never reveals itself in memory or on disk. Standard crypto implementations leave both the algorithm and key vulnerable to tampering and reverse engineering. WhiteboxCRYPTO™ mathematically transforms the key into a complex graph of numbers and executable code. This graph has multiple valid paths ran-domly chosen at runtime based on a user-supplied random source.

Hardware ID binding allows integration of a hardware identifier into the WhiteboxCRYPTO™ library forcing an attacker to reverse engineer a complex, dynamically changing key-graph tied to a single hardware system.

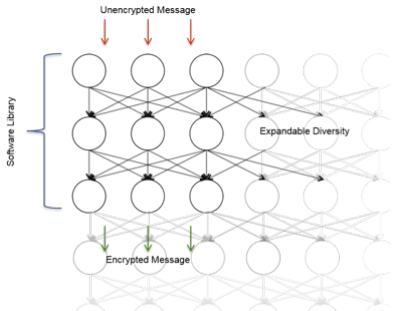

Figure 1: WhiteboxCRYPTO™ transforms the key into a complex key graph of numbers and executable code using mathematical decomposition.

The WhiteboxCRYPTO™ product allows you to: Generate a unique crypto library for each application shipped, and encode the same classical key for each; produce a single library and encode many keys to work with it; or encode multiple keys for multiple libraries. By simply supplying a classical crypto key, tweaking a few configuration variables, and running WhiteboxCRYPTO™ in any build environment, a new library will be generated which can be integrated into an application using a fully documented software API.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo, CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
email: sales.support@microsemi.com
www.microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,400 employees globally. Learn more at www.microsemi.com.

WhiteboxCRYPTO™

Cryptographic key hiding with tunable security and performance

Supported Platforms

WhiteboxCRYPTO™ libraries support configurable key sizes, are little and big endian compatible, run on both 32 bit and 64 bit systems, and are fully compatible with any environment that can link C libraries. WhiteboxCRYPTO™ was developed in a US-only facility by cleared US citizens, is EAR export controlled, and is immediately ready for deployment in C and C++ software applications running on nearly any hardware and operating system configuration.

Use Cases

WhiteboxCRYPTO™ is useful wherever cryptography must be performed in a potentially vulnerable environment or where the crypto keys and/or plaintext data must be protected even if an untrusted user has taken complete control of the host system. Such use cases include compromise of networked systems, software delivered to business competitors, or commercially deployed software with private keys.

Additionally, WhiteboxCRYPTO™ can receive input and produce output in an obfuscated data format, suitable for use with other algorithms in the WhiteboxCRYPTO™ suite. In this way, WhiteboxCRYPTO™ can keep data secured in addition to key material.

Security Features

- Completely Hides Keys
- Bind Keys to Hardware
- Runtime Randomization
- Customize Performance
- Fortify with CodeSEAL™

Features and Benefits

Features	WhiteboxCRYPTO™ Benefits
Hides Keys Completely	Actual key bits never form in memory, thwarting various memory attacks. The obfuscated white box form is resistant to break-once-run-everywhere exploits.
Simple, Documented API	A simple, fully documented API enables quick implementation of secure encryption, decryption, signing, and verifying functionality.
Binds Keys to Hardware	Hardware identifiers can be mathematically integrated into a key, binding an application and sensitive data to a particular hardware platform.
Customizable Performance	Tunable encrypt/decrypt throughput allows full performance vs. security tradeoffs.
Randomizes at Runtime	For each execution, WhiteboxCRYPTO™ takes a random path, confusing and forcing an attacker to reverse engineer a complex node graph.
Highly Portable	Source code based implementation is portable to all platforms and compatible with any software protection technique. WhiteboxCRYPTO functions as little or big endian, 32 or 64-bit, compatible with any environment that can link C libraries.
Managed Keys Solution	The white box version of the key can be stored externally to a WhiteboxCRYPTO™ library enabling key updates, key escrow, etc.
Protects data in-transit	WhiteboxCRYPTO™ produces data in obfuscated form usable by other algorithms within the WhiteboxCRYPTO™ suite. Thus, data is protected during intermediate stages of a sequence of cryptographic operations