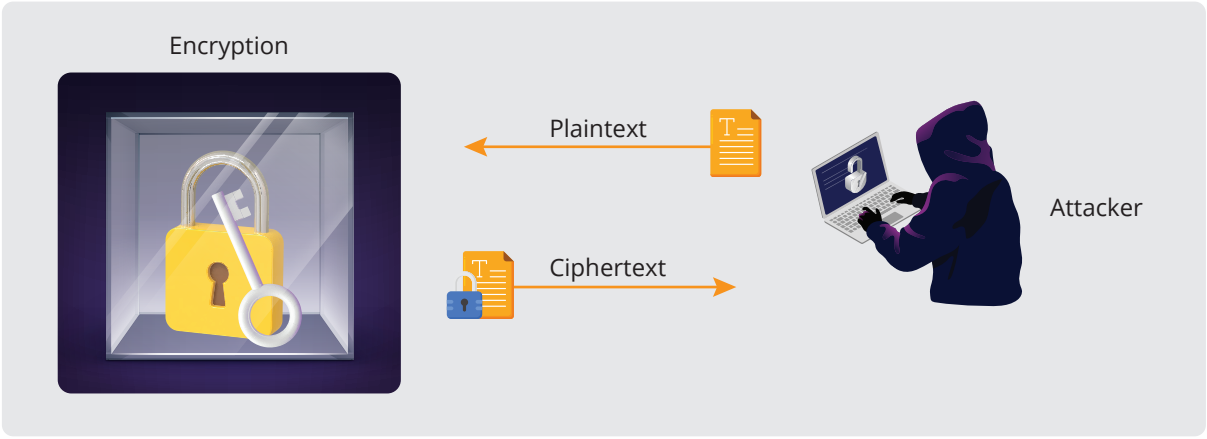
Whitepaper

WHITEBOX CRYPTOGRAPHY

appsealing

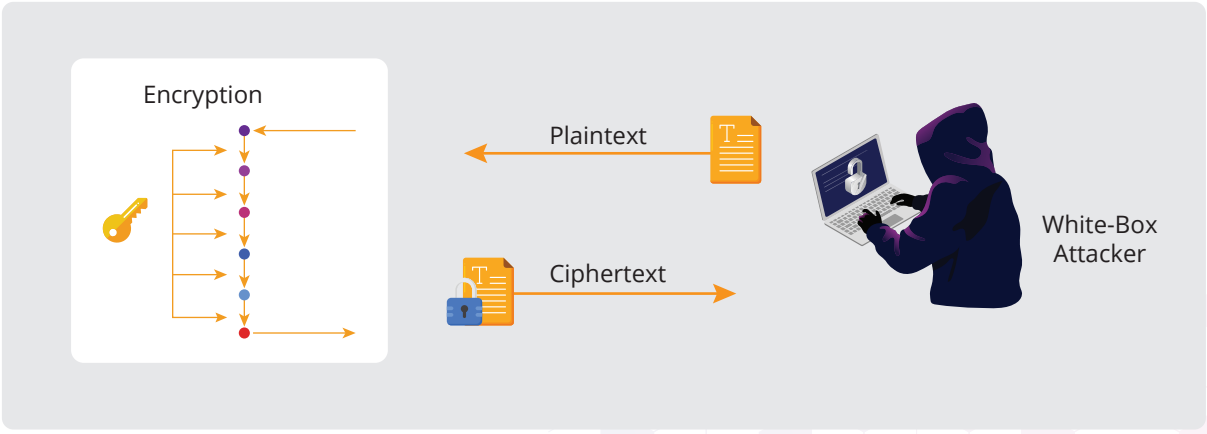
What is White-Box Cryptography?

In the world of encryption, attackers often try to figure out part or all of the encryption keys when they know both the input and output of the encryption process. Block ciphers like AES and LEA have been closely examined to make sure they can combat these kinds of attacks.

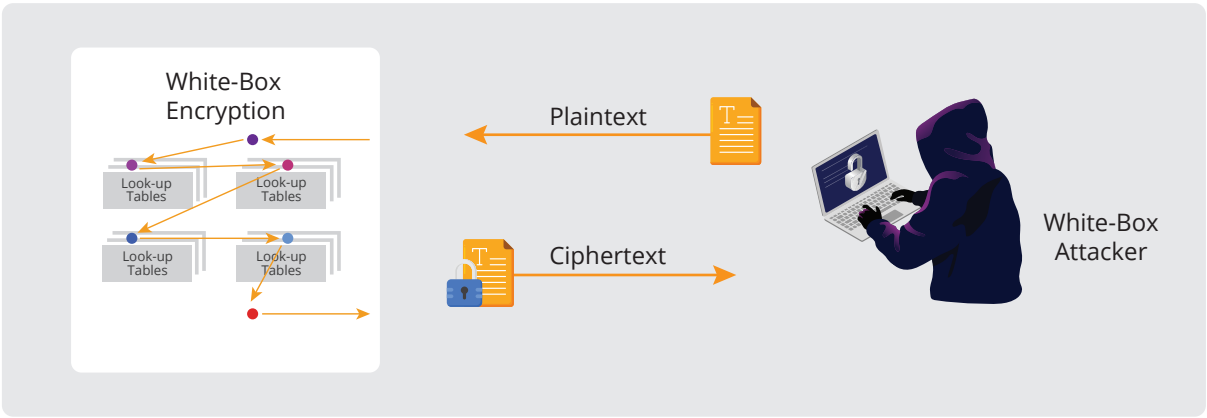


But, if a user or a device gets compromised by bad actors, attackers might peek into the memory space where the encryption or decryption is happening. In the worst-case scenario, attackers can use a debugger to closely examine the specific steps in the instructions and see the changes in the memory.

When attackers know everything about the values in the encryption process, and try to figure out part or all of the key, it is called a white-box attack. Standard block ciphers, without any special features, are weak against white-box attacks, letting attackers quickly grab the encryption key from memory.



On the flip side, white-box cryptography means using ciphers that are specially built to stop the key from being exposed, even if an attacker is in the best possible position. For instance, one common trick is spreading the encryption key across big tables, making it tough for an attacker to guess the key just by looking at those tables in the memory. So, in summary, white-box cryptography achieves its goal by concealing the encryption key in various ways, making it a form of key obfuscation.



White-Box Cryptography Technology of AppSealing

AppSealing prevents abnormal execution of the app, such as running in a debugger environment. However, more fundamentally, white-box cryptography is used to prevent the exposure of keys for encrypting or decrypting crucial data even in situations where white-box attacks are possible.

While various papers have proposed methods to implement the standard block cipher AES as a white-box cryptography, all of them have been susceptible to attack methods. Alternatively, AppSealing has implemented the standard block cipher LEA as a white-box cryptography through modification.

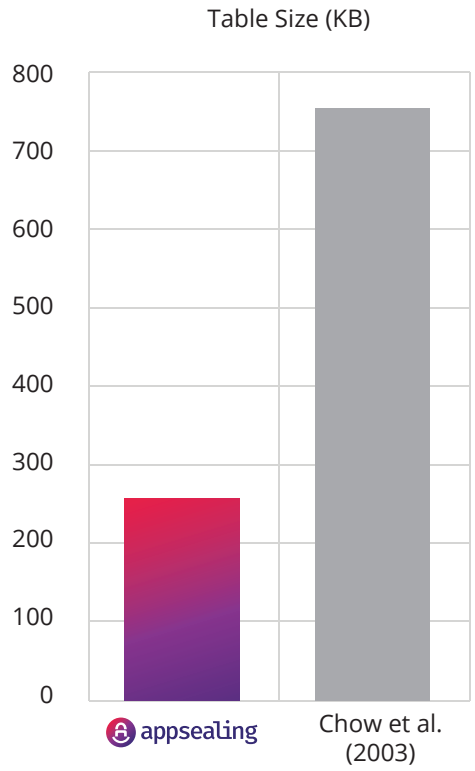
AppSealing's white-box cryptography has the following features:

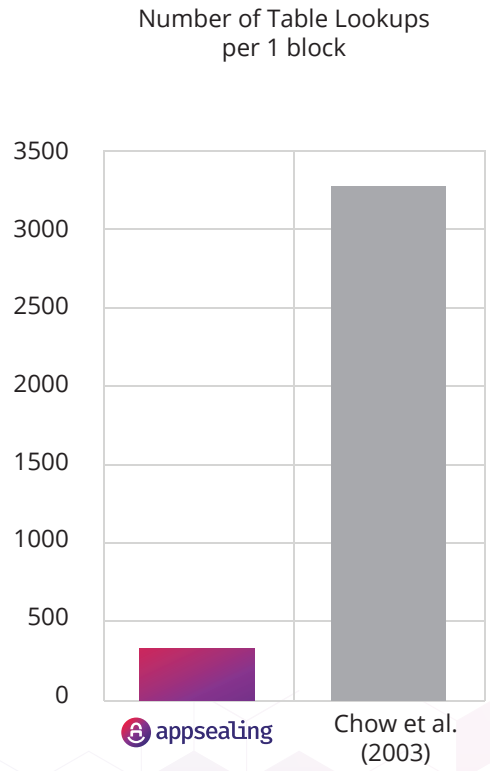
Efficient Memory Usage and High Encryption Speed

Many white-box cryptography implementations adopt a key dispersion method that relies on extensive tables, resulting in substantial memory consumption. Additionally, the frequent referencing of tables in such implementations can lead to a decline in performance. For instance, the method proposed by Chow et al. (2003), a notable white-box cryptography implementation for AES, employs approximately 750 kB of tables and requires over 3,000 table lookups to encrypt a single block.

In contrast, AppSealing's white-box cryptography has undergone multiple memory optimizations, utilizing only about 240 kB of tables and involving fewer than 250 table references. Furthermore, optimizations at the architectural level reduce the impact of cache misses that typically occur when referencing large tables.

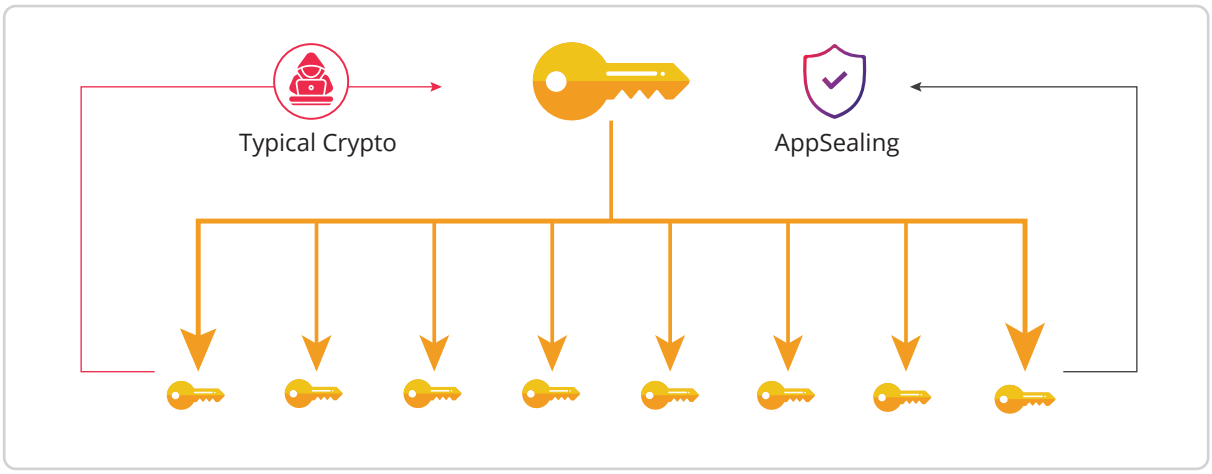
The exceptional performance of AppSealing's white-box cryptography ensures quick operation of various AppSealing features, minimizing any adverse effects on the app's execution.





Resistance to Attack Propagation

Standard block ciphers like AES or LEA typically involve multiple rounds, with the key contributing to the derivation of distinct round keys for each round. Consequently, the exposure of one round key can result in gaining partial information about the entire key. This common vulnerability extends to white-box ciphers directly implemented from standard block ciphers.



In AppSealing's white-box cryptography, LEA undergoes a transformation wherein each round key is generated through a one-way function. Even if certain round keys are exposed, it becomes infeasible to deduce information about the original key from them. Therefore, compromising a specific round key does not immediately translate to vulnerabilities in other round keys.

Enhanced Protection for Look-Up Tables

In AppSealing's white-box cryptography, we incorporate additional internal safeguards before and after utilizing tables. Just as plaintext undergoes a complex transformation to become ciphertext, the tables themselves undergo shuffling and recombination. This intricate design poses a challenge for attackers, even if they manage to access the binary of the sealed app.

How Does AppSealing Use White-Box Cryptography?

AppSealing uses white-box cryptography in various ways to protect applications.

1. Code Encryption

The app's code is encrypted using white-box cryptography by AppSealing. Even if an adversary manages to obtain a code dump, the encrypted code remains resistant to reverse engineering.

2. Data Encryption

As per customer preferences, critical data can be encrypted using AppSealing's white-box cryptography. This is particularly useful when safeguarding sensitive strings, such as secret keys embedded in the code, preventing exposure.
Attackers are unable to decipher essential strings from the binary.

3. External Module Encryption

Upon request, AppSealing can encrypt dynamically required libraries used by the app. Libraries containing functionalities utilized by AppSealing is also protected by white-box cryptography.

Moreover, AppSealing's features enable the generation of credentials for verification on the login server. White-box cryptography is employed during the device's credential generation process.

About AppSealing

With better visibility & insight comes better protection. AppSealing is a trusted player in the world of mobile application security. In today's application-focused world, security should not slow down your speed of development. We utilize runtime application self-protection features to build scalable security solutions for your mobile apps business in quick time **without "ANY CODING".**Our powerful security suite ensures real-time in-depth application security like source code protection, anti-reverse engineering, cheat tool & emulator detection/blocking, and enforces app integrity.
It protects 800+ mobile apps and 800 million+ devices, successfully blocking 70 million+ threats across the globe. Our esteemed clientele spans across Gaming, Fintech, Movie apps, E-comm, Healthcare, and 020.

Writer
Geo Yang
geo.yang@inka.co.kr
White-box & iOS SDK engineer