Zimperium zKeyBox

Secure Your Keys,
Safeguard Your Data

ZIMPERIUM®

How Zimperium Can Help Secure Keys

Zimperium zKeyBox leverages white-box cryptography to protect the cryptographic keys used within your mobile applications. This solution provides a white box-protected cryptographic library for executing all cryptographic operations within applications while running on a mobile device. The main purpose of zKeyBox is to ensure that cryptographic keys are never revealed in plain text when they are at rest, in motion, or even when in use. With such security in place, it becomes challenging for attackers to locate, modify, and extract cryptographic keys even when the device is under the attacker's control.

The white-box transformed keys are generated on-premise, so **Zimperium never sees your keys.**

Key Benefits

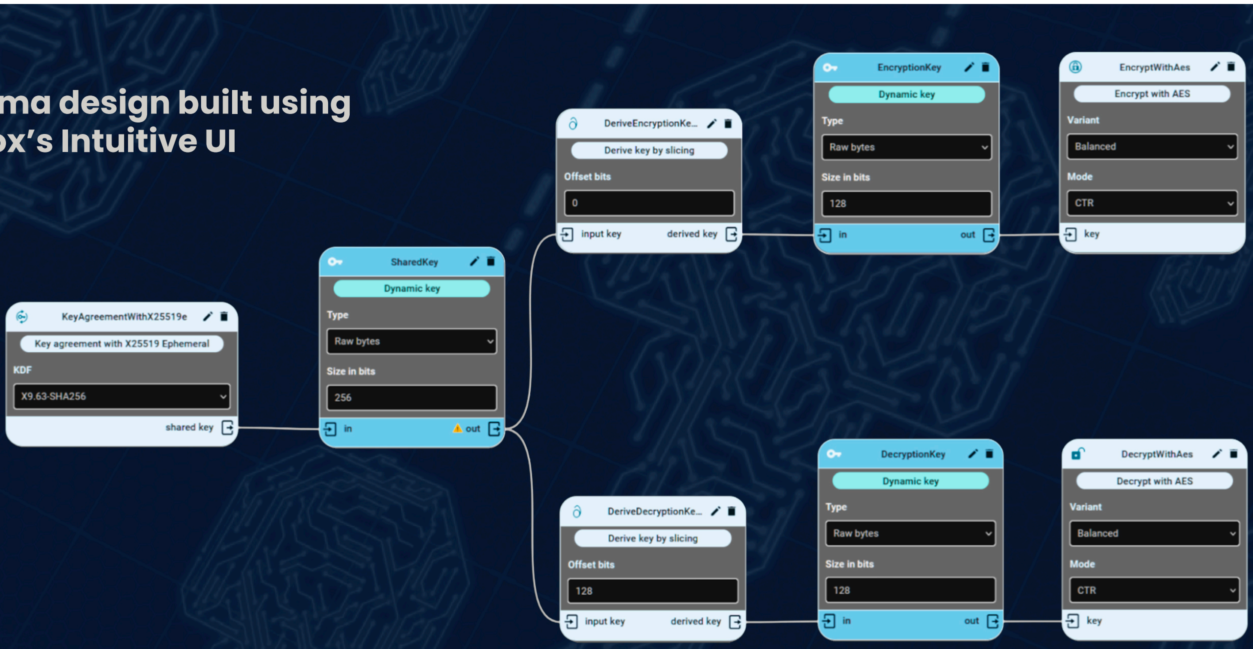
1. **Embed Keys With Confidence:** Once keys are protected, they can be embedded within your application code.
2. **Enhanced Security in Untrusted Environments:** It protects keys even if the device is compromised, making it ideal for mobile applications that may run on insecure or compromised devices.
3. **Platform-Independent:** White-box cryptography offers robust protection without relying on hardware-based security mechanisms, such as Secure Enclave, Android Keystore, TPM modules, or other Trusted Execution Environments (TEEs).
4. **Same Level of Security Across All Platforms:** Ensures the same level of security on mobile and non-mobile platforms regardless of the OS version.

The average cost of a data breach rose to **$4.24 million** in 2021 globally based on the 2021 IBM Data Breach Report. And these don't even take into account mega breaches where that cost jumps exponentially to **$401 million**.

Why Zimperium's zKeyBox

Support Many Standard & Custom Algorithms	Protect any cryptographic scheme using algorithms such as AES, 3DES, RSA, ECC, HMAC, etc. Custom algorithm support is also available.
Comprehensive Platform Support	Linux (glibc, uClibc, musl), Windows, macOS, Android, iOS, tvOS, watchOS, Xbox, PlayStation, WebAssembly, and others.
Design The Right Schema	Visual builder guides the user by providing precise warnings and errors for invalid or suboptimal conditions, like when the key sizes don't match the chosen algorithm.
Auto Generate Code From Schema	Unlike traditional solutions, our graphical interface simplifies schema design and automatically generates code for the white-box library, accelerating implementation and reducing manual coding efforts.
Built-In Support for Securing PIN Data	zKeyBox undergoes regular penetration testing, supports DUKPT key management, TR-31 key blocks, standards compliant random generation with reseeding capability and other features that help obtain various certifications for the protected application and separation of payment card and PIN data as specified by PCI-DSS.
Security Architecture Guidance	We provide guidance on the best security architecture and requirements to ensure optimal implementation for your specific needs.
FIPS 140-3 Certification	Cryptographic modules developed by this solution meet the stringent requirements of FIPS 140-3 Level 1.
Easy Integration	Plug-and-play replacement for standard cryptographic libraries. Easy to use APIs as there are no excessive parameters to specify.

A schema design built using zKeyBox's Intuitive UI



KeyAgreementWithX25519e
Key agreement with X25519 Ephemeral
KDF: X9.63-SHA256
shared key

SharedKey
Dynamic key
Type: Raw bytes
Size in bits: 256
in
out

DeriveEncryptionKe...
Derive key by slicing
Offset bits: 0
input key
derived key

EncryptionKey
Dynamic key
Type: Raw bytes
Size in bits: 128
in
out

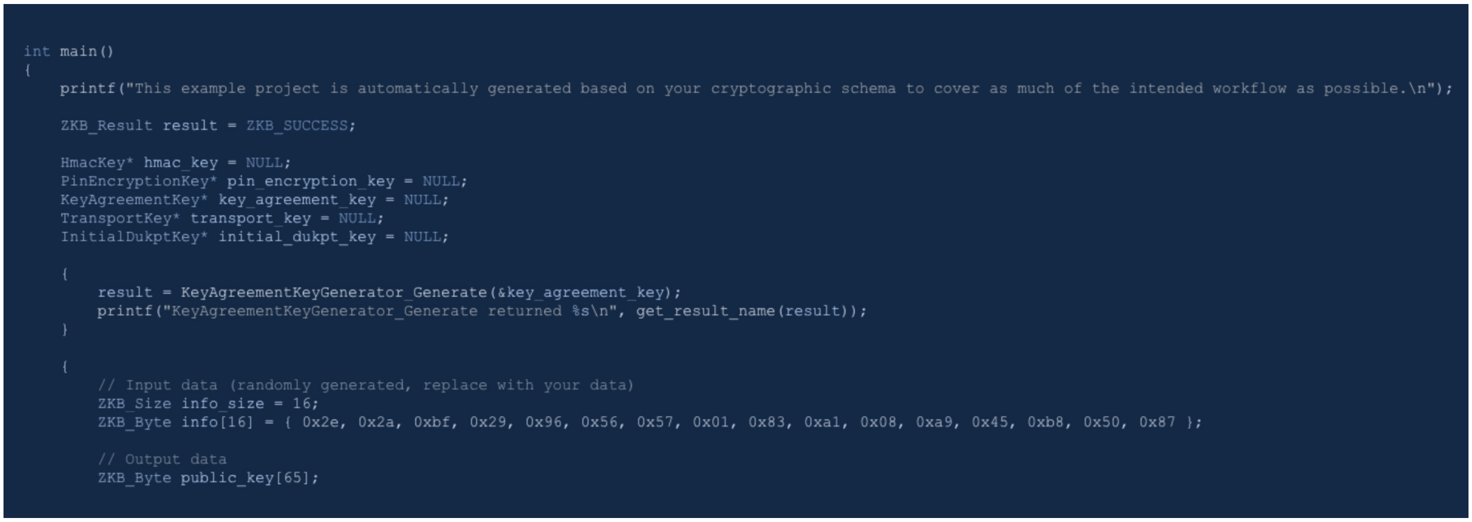
EncryptWithAes
Encrypt with AES
Variant: Balanced
Mode: CTR
key

DeriveDecryptionKe...
Derive key by slicing
Offset bits: 128
input key
derived key

DecryptionKey
Dynamic key
Type: Raw bytes
Size in bits: 128
in
out

DecryptWithAes
Decrypt with AES
Variant: Balanced
Mode: CTR
key

Using the schema design, code is automatically generated



```
int main()
{
    printf("This example project is automatically generated based on your cryptographic schema to cover as much of the intended workflow as possible.\n");

    ZKB_Result result = ZKB_SUCCESS;

    HmacKey* hmac_key = NULL;
    PinEncryptionKey* pin_encryption_key = NULL;
    KeyAgreementKey* key_agreement_key = NULL;
    TransportKey* transport_key = NULL;
    InitialDukptKey* initial_dukpt_key = NULL;

    {
        result = KeyAgreementKeyGenerator_Generate(&key_agreement_key);
        printf("KeyAgreementKeyGenerator_Generate returned %s\n", get_result_name(result));
    }

    {
        // Input data (randomly generated, replace with your data)
        ZKB_Size info_size = 16;
        ZKB_Byte info[16] = { 0x2e, 0x2a, 0xbf, 0x29, 0x96, 0x56, 0x57, 0x01, 0x83, 0xa1, 0x08, 0xa9, 0x45, 0xb8, 0x50, 0x87 };

        // Output data
        ZKB_Byte public_key[65];
```

Easy to Implement

- **Step 1:** Design the cryptographic schema using Visual Builder.
- **Step 2:** Generate the zKeybox package, which includes a library and tools.
- **Step 3**: Transform your plain keys into protected keys using the tools.
- **Step 4:** Integrate the zKeybox library into your mobile application.
- **Step 5:** Import the protected keys into the zKeyBox library.
- **Step 6:** Call the zKeyBox library to perform a cryptographic function.

How Customers are Using our Solution

Contactless Payments on Mobile Phones

Mobile phones are increasingly used as contactless payment terminals. Developing payment software for general-purpose phones is significantly cheaper and more convenient than creating specialized hardware for traditional point-of-sale systems. The danger lies in that software-based security systems are easier to reverse engineer than special-purpose hardware. By extracting the internal cryptographic keys, an adversary can collect sensitive data, steal money, or disrupt business operations. The industry-standard approach to mitigating key extraction risks on general-purpose devices is white-box cryptography. In fact, the Payment Card Industry Security Standards Council (PCI SSC) requires all vendors to employ white-box cryptography for mobile contactless payment applications to protect keys. zKeyBox is a white-box cryptography library, which means that by using it in a software-based payment system you are able to satisfy the PCI SSC security and testing requirements and ensure strong security.

Reduce Content Piracy

Streaming providers leverage multiple crypto key algorithms in a layered fashion to encrypt the content and entitlements of the subscribers. They rely on secure chip hardware to store the content decryption keys within the set-top box. However, most providers use set-top-box (STB) hardware from several manufacturers who don't always support the encryption of their choice, making the content vulnerable to piracy. Customers are leveraging zKeyBox to protect content on set-top boxes with incompatible hardware in the short term. But in the long run, they plan to completely untether themselves from hardware-based security to save high refresh costs and reduce the risk of piracy.

Global digital piracy costs US film and TV industry at least an estimated

$29.2 Billion

and as much as $71 billion annually, according to a new study from the US Chamber of Commerce's Global Innovation Policy Center.

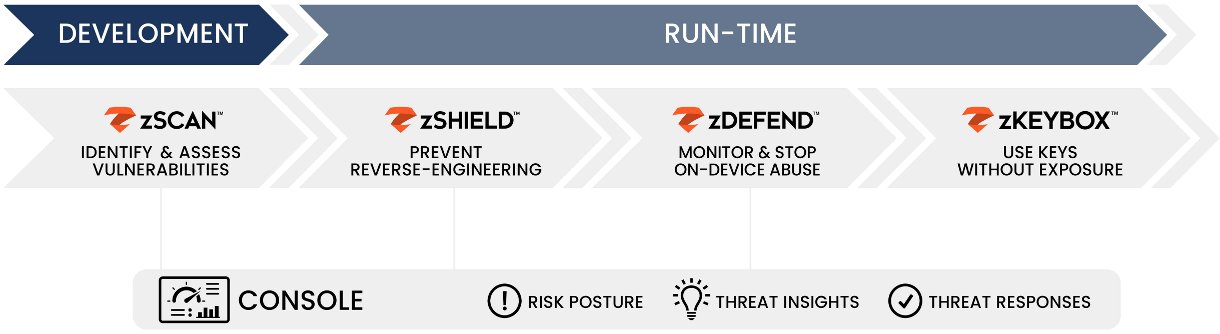
Bring Your Own Key (BYOK)

Customers migrate enterprise applications that were traditionally on-premise to the cloud. But, they do not trust cloud providers with their cloud data encryption keys due to the extent of provider access, limiting encryption options, and lack of control over the key management lifecycle. They are choosing to keep the key generation and management on-premise but leverage zKeyBox's white-box cryptography to secure all cryptographic activities carried out by the application.

Why Zimperium MAPS

Zimperium's Mobile Application Protection Suite (MAPS) helps enterprises build safe and secure mobile applications resistant to attacks. It is the only unified solution that combines comprehensive application protection with centralized threat visibility.

MAPS comprises four capabilities, each of which address a specific enterprise need as shown below.



DEVELOPMENT

RUN-TIME

zSCAN™
IDENTIFY & ASSESS VULNERABILITIES

zSHIELD™
PREVENT REVERSE-ENGINEERING

zDEFEND™
MONITOR & STOP ON-DEVICE ABUSE

zKEYBOX™
USE KEYS WITHOUT EXPOSURE

CONSOLE

RISK POSTURE

THREAT INSIGHTS

THREAT RESPONSES

Organizations with a high level of cloud migration had an average cost of a breach of

$5.12 Million

compared to $3.46 million for organizations with low levels of cloud migration, for a difference of $1.66 million or

38.7%.

Solutions	Value Proposition
zSCAN™	Helps organizations continuously discover and fix compliance, privacy, and security issues prior to being published.
zSHIELD™	Protects the source code, intellectual property (IP), and data from potential attacks like reverse engineering and code tampering.
zDEFEND™	Provides threat visibility and on-device ML-based run-time protection against device, network, phishing, and malware attacks.
zKEYBOX™	Protect your keys so they cannot be discovered, extracted, or manipulated.

PROTECT YOUR KEYS TODAY

We can ensure the security of your most sensitive data regardless of the hardware your mobile application runs on. Contact us for more information.

ZIMPERIUM®

Learn more at: zimperium.com
Contact us at: 844.601.6760 | info@zimperium.com
Zimperium, Inc
4055 Valley View, Dallas, TX 75244