Mobile apps emulating plastic cards now encompass everything from payment, banking, healthcare, and access control and transit. These apps now reside on a wide range of devices, such as phones, tablets, key fobs, watches and other wearables. As a result, sensitive data, and functionality are increasingly widely distributed and are therefore exposed to greater potential for theft and subversion. Securing this wide and growing ecosystem or interconnected mobile apps requires much more than just traditional standard data encryption. Developers must build comprehensive protection into app content, logic, data, and cryptography. The solution to these problems is **MatrixSSE™**.

**INSIDE MatrixSSE™** provides secure element functionality in software. It provides powerful protections for apps

- Secure processing: **MatrixSSE™** WhiteBox component prevents secrets from being revealed in memory while they are being used by processing them behind a cryptographic boundary. **MatrixSSE™** also protects against any attacks on weaknesses in the original cryptographic algorithm. WhiteBox is a cryptographic solution that allows an application to perform cryptographic operations without exposing keys and other secrets that are critical to the security of those operations. It ensures that these critical, secret elements are never revealed even while the cryptographic operations are being executed.

- Tamper resistance: This technology enables **MatrixSSE™** (CORE component)-protected applications to defend themselves with a comprehensive runtime integrity network, preventing analysis, debugging, subversion, malware, and other forms of attack from influencing app operations. A **MatrixSSE™** protected software program constantly checks itself for health as it runs, identifying even a single bit or byte out of place. Depending on needs, the integrity network can attempt to repair any problems it finds, report to a pre-programmed location, or terminate the execution of the program.

- Obfuscation: **MatrixSSE™** (CONCEALER component) obscures app operations so that hackers cannot understand the app or its logic. Obfuscation is useful for protecting jailbreak detection and other critical security functions (including 3rd party components) from reverse engineering.

In **INSIDE MatrixSSE™** product advanced heuristics automatically apply security, so that specialist security and development teams are not required for production implementation. **MatrixSSE™** integrates seamlessly with existing build processes and scales to multiple development projects, enabling hundreds of apps to be protected across any organization.

**crypto**
- cryptographic support
  - AES encrypt & decrypt (all key sizes)
  - RSA (PKCS#1) encrypt, decrypt & sign (all key sizes)
  - ECC-derived algorithms, including ECDSA (all key sizes)
  - Elgamal
  - Diffie-Hellman
  - HMAC
  - OMAC

**markets**
- HCE payment
- Healthcare
- Transit
- Access control
- Automotive

**platforms**
- iOS
- Android
- Blackberry 10
Gain comprehensive protection

MatrixSSE™ is an HCE security tool kit that stops App hacking, including attacks from high-grade, corporate, or nation-states. It protects code (payment logic and mechanisms), data (tokens, sensible credentials, security parameters and counters), and communications (data or token provision) end-to-end across the software and cloud services. This security level is applied at source code level and not at executable level in order to embed security as deeply into then App Code as possible and also to limit impact on app performances, meeting transaction timing requirements from service providers.

Create trust with secure provisioning and communication

MatrixSSE™ can authenticate multiple system components such as another application, process, hardware device, or server. This ensures that the protected software is communicating with the intended recipient and not a proxy or replacement component. Checks are performed non-deterministically without affecting app performances. Hackers have no way of understanding when authentication takes place or what form it takes. This makes it impractical for the hacker to spoof or attack. With MatrixSSE™, you can create trust frameworks between different systems and define how to respond to attacks.

Process data securely

Protect sensitive data with MatrixSSE™ with full control of encryption, decryption, and processing. MatrixSSE™ white box cryptography prevents private keys being read in memory and cryptographic algorithm operation from being understood. Data is processed behind a cryptographic obfuscation boundary within the application at its point of need, then discarded or re-encrypted as needed. MatrixSSE™ uniquely provides context enforcement between the program and the cryptographic implementation. The cryptographic operation must be executed in the correct context and state of the program. If it is used in a different context, such as being lifted or repurposed, then all results will be incorrect. This makes it impractical to use the encryption out of context or to call the encryption routines in place in an attempt to subvert them.

Obfuscate application code

Obfuscation technology prevents logic and sensitive functions from being reverse-engineered. You can obfuscate an entire app, key functions, or both, and protect them from static and dynamic analysis. Gain full control over obfuscation that is created in layers with multiple passes through the application. Obfuscation is used in conjunction with app Integrity checking (anti-tamper protection) to provide global security at system level.

Detect threats in real time

MatrixSSE™ instantly detects application hacking attempts in real time. It uses a network of many thousands of individual “checks” that are intricately interconnected and cover the entire binary. The check network is entirely interrelated, meaning that attacking individual checks will result in the whole network detecting the attack. This complex coverage ensures that even changing one bit of application code or static data will be detected, allowing tamper policies to be aligned with security and risk management requirements.

For further details on all of INSIDE’s security solutions, visit www.insidesecure.com

---

What is WhiteBox Cryptography?

WhiteBox cryptography prevents private keys from being read in memory and cryptographic algorithm operation from being understood. Data is processed behind a cryptographic obfuscation boundary within the application at its point of need, then discarded or re-encrypted as needed. This makes it impractical to use the encryption out of context or to call the encryption routines in place in an attempt to subvert them.

---

MatrixSSE™ integration

This diagram illustrates the integration of MatrixSSE™ with various components such as Mobile App, Native Code, Java, Sensitive data, Communication through WhiteBox crypto, DAR (Data At Rest) through WhiteBox encryption, and more.