An overview of Zebra's Secure Element (SE) and Secure Access Module (SAM) technologies

Enterprise-grade security for a trusted future

Introduction: The growing need for hardware-based security

In today's mobile-first enterprise environment, an increasing amount of sensitive data is handled on mobile devices, from financial transactions and personally identifiable information (PII) to private corporate data. While software-based encryption offers a layer of protection, it is not immune to sophisticated attacks. For true, robust security, the data and the cryptographic keys that protect it must be shielded in a dedicated, tamper-resistant hardware environment.

Zebra Technologies recognizes that secure key management is the cornerstone of mobile and enterprise security. To address this critical need, Zebra has integrated advanced hardware security solutions into its enterprise devices. This document provides a high-level overview of Zebra's implementation of the Secure Element (SE), its foundation in Google's Android StrongBox solution, and the SAM functionality. It explains the value these technologies deliver to customers and highlights the Zebra products that feature SE/TEE (Trusted Execution Environment, a secure area within the main processor)/SAM.

Note: For key similarities and differences between SE and TEE featured on Zebra products please refer to Zebra's white paper "Enterprise-Grade Security for a Trusted Future. Zebra Secure Element"

What is a Secure Element (SE)?

A SE is a dedicated, tamper-resistant microchip with its own processor and secure storage, physically isolated from the device's main processor. Its primary function is to act as a digital vault, protecting sensitive data, like cryptographic keys, from software-level attacks that could compromise the main operating system. It is a small, self-contained computer within the device that is highly isolated from the main operating system.

Key characteristics of a Secure Element include:

- Dedicated hardware: It has its own CPU, secure storage, and RAM.
- High security: It is designed to be resistant to both physical and software attacks, including side-channel attacks.
- Isolation: It runs independently of the main Android™ OS, so even if the main operating system is compromised, the data and keys within the SE remain protected.

The Android ecosystem officially supports this technology through Google's Android StrongBox, which is an implementation of its security module within a dedicated, tamper-resistant hardware component like a Secure Element. As defined by Google, a StrongBox implementation includes:

- Its own dedicated CPU and secure storage.
- A true random-number generator for creating strong cryptographic keys.
- Mechanisms to resist physical tempering.
- The ability to prevent the extraction of private keys.

By using the Android Keystore system to interact with the StrongBox SE, applications can perform cryptographic operations where the key material is never exposed outside the secure hardware. Even if the Android operating system were compromised, an attacker could not extract these hardware-bound keys.

ZEBRA

The Zebra Secure Element implementation: Powered by Google StrongBox

Zebra's Secure Element implementation is based on the industry-recognized Google Android StrongBox standard. This demonstrates Zebra's commitment to providing best-in-class, enterprise-grade security that aligns with a trusted and standardized ecosystem. This integration provides a level of security against both software attacks and physical tampering that software-only solutions cannot match.

This advanced security feature is already implemented and proven in a range of key Zebra products, including the FR55, TC53e, TC58e, TC501, TC701, and ET401.

Note: For the most current list of Zebra products supporting SE, please contact the Zebra Support Center or consult official product documentation.

Practical applications for the SE

The true power of the Secure Element is realized in its practical applications.

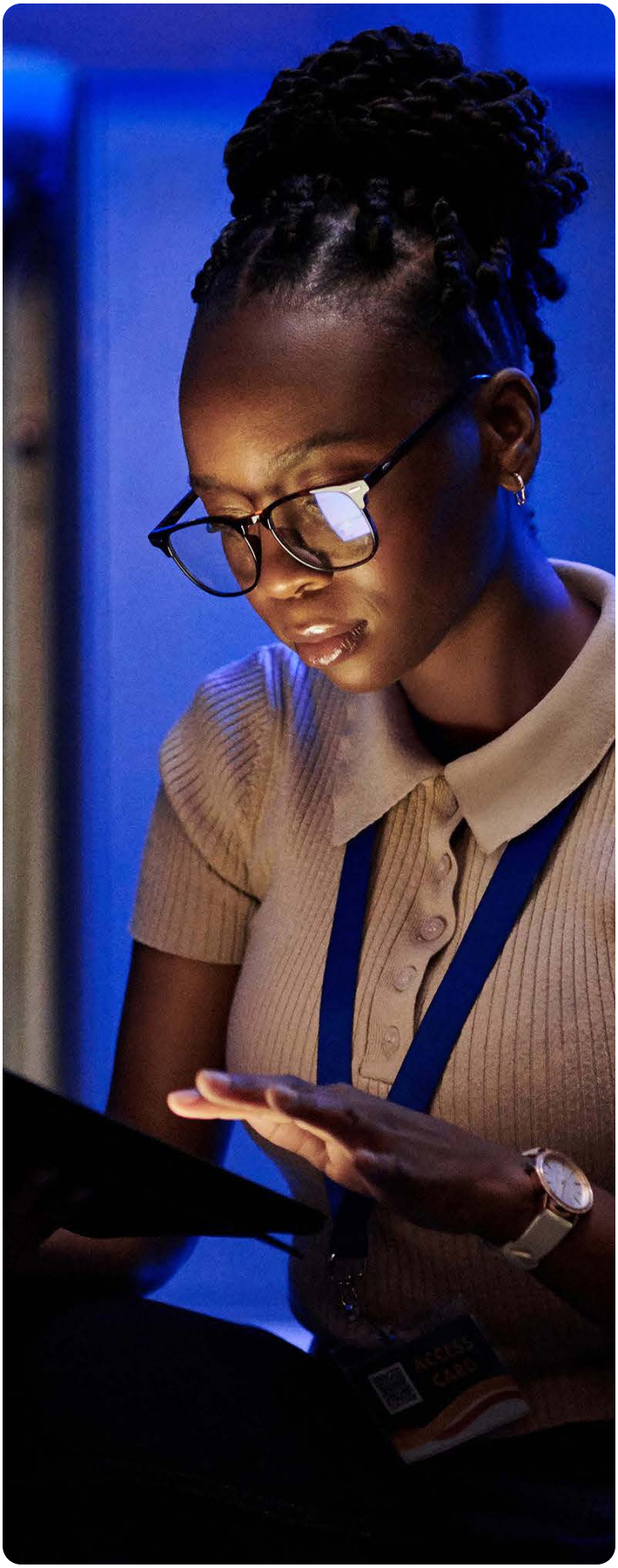
Use case: Secure validation of Apple and Google Wallet passes

- Scenario: A customer at a venue or a guest at a hotel presents a digital event pass or room key stored in their Apple or Google Wallet.
- The SE solution: The certificates and private keys required for the cryptographic validation of the pass are stored within the Zebra device's Secure Element. The entire validation process occurs inside this secure chip, and the private key never leaves the SE. This ensures the transaction is secure and impervious to software-based snooping, allowing for a frictionless yet highly secure customer experience.

Use case: Protecting device and application integrity with key attestation

- Scenario: An application needs to access sensitive corporate resources from a remote server, which must verify that it is communicating with a genuine, uncompromised Zebra device.
- The SE Solution: The Secure Element supports key attestation, which cryptographically proves that a key used by an application is protected by hardware. This allows the server to trust that it is connected to a legitimate device, preventing unauthorized access from emulated or compromised devices and strengthening the overall security posture of the enterprise network.

Note: Both use cases mentioned for SE are also fully supported with a TEE. The similarities and differences between SE and TEE are described in the "Enterprise-Grade Security for a Trusted Future—Zebra Secure Element" white paper.



What is a Secure Access Module (SAM)?

A SAM is a small (SIM-like), secure hardware component, typically in the form of a smart card housed in a dedicated slot (similar to a SIM card slot). Its primary purpose is to provide security services for specific applications by securely storing cryptographic keys and performing cryptographic operations.

Several predominant SAM technologies are in use, including MIFARE SAM from NXP, FeliCa SAM from Sony, Calypso SAM from the Calypso Network Association/Innovatron, and ITSO from the Integrated Transport Smartcard Organization in the UK.

Unlike an embedded Secure Element that provides broad, OS-level security, a SAM is often used to enable secure transactions with external systems or services. This is particularly common in industries requiring a dedicated, often third-party-managed, security token. The Zebra FR55 is an example of a product that incorporates SAM technology, providing a dedicated slot to enable enhanced security for specific use cases.

Practical applications for the SAM

SAMs are instrumental in industries that rely on high-assurance security for transactions and authentication.

Industry: Public transportation and tolling

Use Case: A transit authority needs to securely validate contactless fare cards. The SAM in a reader (like the FR55) can be provisioned with the transit authority's master keys. When a passenger taps their card, the SAM performs cryptographic validation, ensuring the card is authentic and preventing fraud without the keys ever being exposed in the reader's software. The SAM also enables dedicated secure data operations during the transit fare transaction validation.

Industry: Retail and payment systems

Use Case: In closed-loop payment systems (e.g., a corporate campus or stadium), a SAM can be used to securely authorize and process payments from proprietary cards. The SAM holds the keys necessary to decrypt payment information and validate transactions, isolating these critical functions from the main point-of-sale software.

Similarities and differences: Android StrongBox vs. SAM

While both technologies provide hardware-based security, they differ in their implementation and primary purpose. Both SAM (Secure Access Module) and Google Android StrongBox are hardware-based security solutions designed to protect cryptographic keys and perform sensitive operations. However, they operate in different ecosystems and are designed for different primary purposes.

Similarities

Here are the key similarities between SAM and Android StrongBox:

- Hardware-based security: Both technologies rely on dedicated, tamper-resistant hardware to provide a secure environment for cryptographic operations and key storage. This hardware is separate from the main application processor.
- Secure key storage: The primary function of both is to securely store cryptographic keys, preventing them from being extracted or compromised.
- Cryptographic operations: Both are capable of performing cryptographic operations such as key generation, encryption, and digital signatures within their secure hardware, ensuring that sensitive key material never leaves the protected environment.
- Enhanced security: Both provide a higher level of security than software-only solutions and even Trusted Execution Environments (TEEs) by offering stronger protection against physical attacks.

Differences: General

The main differences between SAM and Android StrongBox lie in their ecosystem, integration, and intended use cases.

Aspect	Secure Access Module (SAM)	Secure Element (SE)
Form Factor	Typically, a removable module or card is inserted into a dedicated slot.	Typically, an integrated chip soldered directly onto the device's mainboard (embedded).
Integration	Loosely coupled; interacts with specific applications that require its services.	Tightly integrated with the device's operating system (e.g., Android Keystore).
Primary Purpose	Transaction-centric security: enabling secure communication and transactions with an external system.	Device-centric security: protecting the integrity of the device, OS, and on-device data.
Provisioning	Keys are often provisioned by a third-party service provider (e.g., a bank, transit authority, etc.) who owns the SAM. SAM cards do not require over-the-air provisioning, which could introduce additional security risk.	Keys are often provisioned at the factory or managed by the device's OS.

Differences: Features

Feature	Secure Access Module (SAM)	Secure Element (SE)
Primary ecosystem	Primarily used in specific industries like public transportation, access control, and ticketing systems.	Integrated into the Android operating system for use by Android applications.
Integration	Typically integrated into terminals, readers, or other dedicated devices to interact with smart cards or other secure elements.	Implemented as a hardware security module within an Android device, accessible to apps via the Android Keystore API that allows apps to use SE hardware.
API and developer access	Accessed through specific libraries and APIs, often defined by standards like CIPURSE™ or Calypso Network Association.	Accessed by Android app developers through the standard Android Keystore API, by specifying that a key should be "StrongBox-backed".
Typical use cases	Securely loading keys into smart cards, mutual authentication between a terminal and a card, and securing transactions in a closed-loop system.	Protecting sensitive user data in Android apps, securing digital wallets and payments, mobile driver's licenses, and other security-critical features on a smartphone.
Governing body/standard	Standards are often set by industry bodies like the OSPT Alliance (for CIPURSE™).	Part of the Android Open-Source Project (AOSP) and promoted by the Google-led Android Ready SE Alliance.
Hardware implementation	A dedicated secure microcontroller that can be integrated into various form factors.	A specific implementation of the Keymaster HAL (Hardware Abstraction Layer) on a SE chip, such as Google's Titan M chip or STMicro STSAF-S320.

Key selling points: positioning SE vs. SAM

Understanding how to position these technologies is key to addressing customer needs effectively.

Position the SE for device and data integrity:

- Core message: "Android-native, best-in-class security for on-device data and modern mobile applications."
- Use when: The customer needs to protect sensitive data stored on the device, ensure application integrity, or support modern use cases like digital wallets (Apple/Google), mobile IDs, and secure remote access.
- Key differentiator: Tightly integrated with the Android OS via the Google Android StrongBox standard, offering robust, OS-level protection against software and physical attacks.

Position the SAM for transactional security:

- Core message: "A dedicated, industry-standard hardware solution for high-assurance, transaction-based systems."
- Use when: The customer operates in a specific vertical like public transportation, tolling, or closed-loop payments that relies on an existing smart card infrastructure (e.g., MIFARE, Calypso, FeliCa, ITSO).
- Key differentiator: Enables secure transactions with external systems using third-party provisioned keys. The removable nature allows the security ownership to remain with the service provider (e.g., the transit authority).

Zebra's unique advantage:

- Core message: "Zebra provides a flexible, multi-layered security platform to meet any enterprise need."
- Why it matters: By offering devices with both embedded SE capabilities and SAM slots, Zebra demonstrates a commitment to comprehensive security. This allows customers to choose the right tool for the job, whether it's securing the device itself with SE or TEE or integrating it into a specialized transactional ecosystem with a SAM.

Conclusion: A commitment to enterprise-grade security

By integrating both hardware-based Secure Elements (powered by Google's Android StrongBox) and offering SAM capabilities, Zebra provides a multi-layered security platform that customers can trust with their most sensitive operations. For Zebra Sales and vendor partners, these features are powerful differentiators. They demonstrate a clear-commitment to delivering secure, reliable, and future-proof enterprise mobility solutions that protect customer data, secure transactions, and build a foundation of trust in an increasingly connected world.

In essence, a SAM is a more transactional secure hardware component often used in dedicated infrastructure like transit terminals, offering rich functionality specific to SAM technologies like Mifare, Calypso, FeliCa or ITSO. In contrast, Google's Android StrongBox is a specific implementation of a hardware security module within the Android ecosystem, designed to provide strong, hardware-backed security for applications running on Android devices. While both are built on the principle of a hardware root of trust, their application and integration are tailored to their respective environments.

→ **Learn more at www.zebra.com/security**

ZEBRA