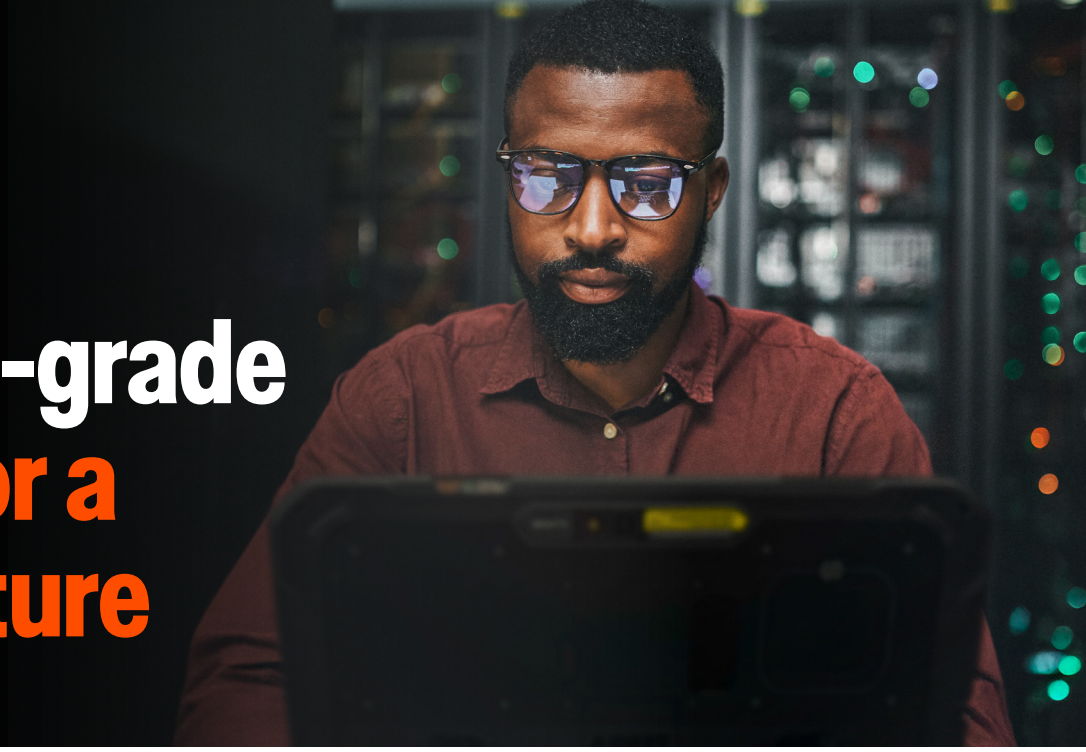


Enterprise-grade security for a trusted future

Zebra secure element—functionality and use cases

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 communications. While software-based encryption offers a layer of protection, it is not immune to sophisticated attacks. For true, robust security, 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 security. To address this critical need, Zebra has integrated a Secure Element (SE) into its enterprise devices. This document provides a high-level overview of Zebra's SE implementation, its foundation in StrongBox, and the value it delivers to customers. Zebra devices that support StrongBox, such as **FR55 Series, TC53e, TC58e, TC501, TC701 and ET401*** utilize the Secure Element for enhanced security, protecting sensitive information against physical tampering and unauthorized access.

What is a Secure Element?

A Secure Element is essentially a dedicated digital vault, physically isolated security chip with its own processor and secure storage. Its primary function is to protect sensitive data, like cryptographic keys, from software-level attacks on the device's main processor.

The Android™ ecosystem officially supports this technology through StrongBox, which is an implementation of its KeyMint security module within a 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 tampering and unauthorized application sideloading.
- The ability to prevent the extraction of private keys and throttle access attempts to prevent brute-force attacks.

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 from the device.

* For the updated list of the Zebra products supporting SE please contact Zebra Support Center.

ZEBRA

The Zebra implementation: powered by StrongBox

Zebra's Secure Element implementation is based on the industry-recognized 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 that software-only solutions cannot match, making Zebra devices hardened against both software attacks and physical tampering.

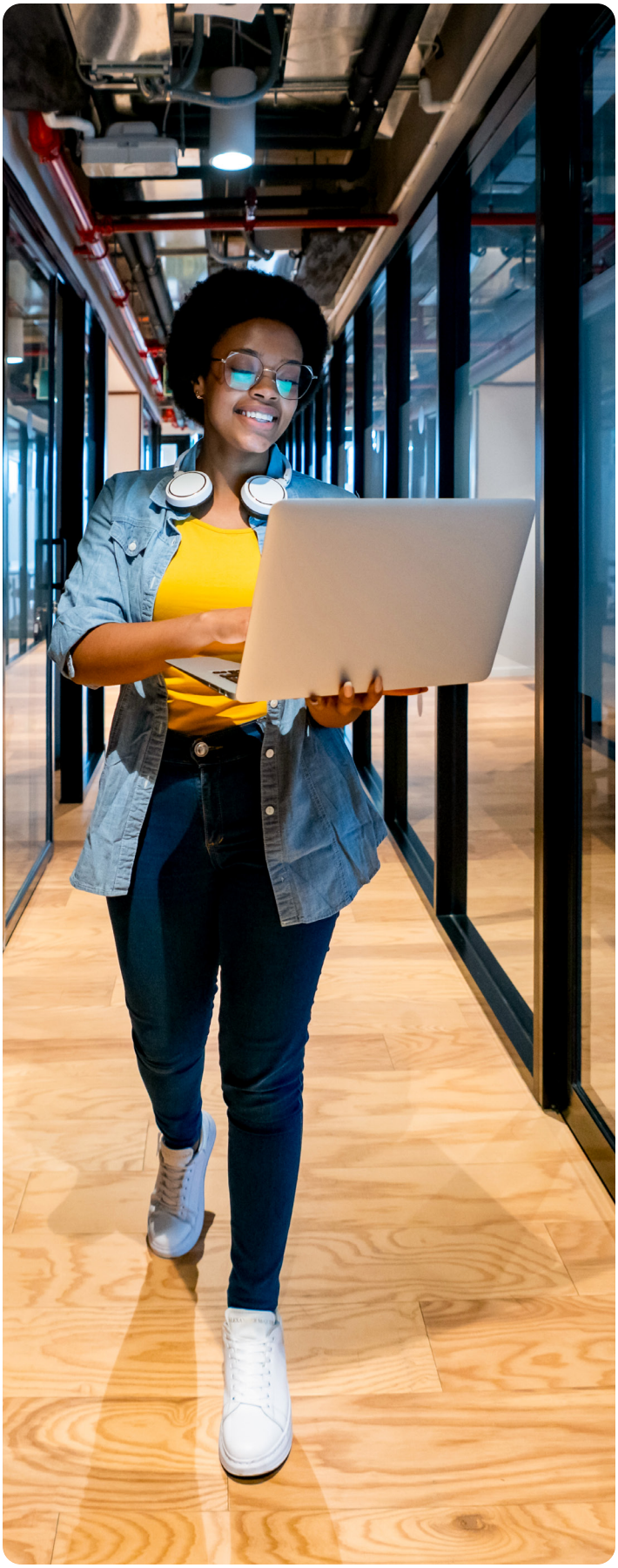
This advanced security feature is already implemented and proven in key Zebra products, including the FR55, TC53e, TC58e, TC501, TC701 and ET401, devices.*

* For the updated list of the Zebra products supporting SE please contact Zebra Support Center.

Key benefits for customers

Translating this technology into customer value is simple. Devices equipped with the Zebra Secure Element offer:

Benefit	What it means for your customers
Enhanced, hardware-based security	The Secure Element is a dedicated security chip, like a digital vault, that is physically isolated from the device's main processor. This makes it extremely resistant to software attacks and malware, providing a level of security that software-only solutions cannot match.
Secure and seamless contactless experiences	By securely storing cryptographic keys, the Secure Element enables Zebra devices to handle sensitive contactless transactions with speed and reliability. This is perfect for use cases like validating Apple & Google VAS passes, ensuring a smooth experience for customers at events, in retail, and beyond.
Future-proofs investment	The Secure Element provides a secure platform to support next-generation applications and services. As new security requirements and use cases emerge, devices with a Secure Element will be ready to adapt, protecting the customer's investment in Zebra technology.
Industry-standard trust	Zebra's implementation is based on the StrongBox KeyMint HAL, an industry-recognized standard for hardware-backed security. This demonstrates Zebra's commitment to providing best-in-class, enterprise-grade security that aligns with a trusted ecosystem.
Proven in Zebra products	You can confidently tell customers that this advanced security feature is already implemented and proven in key Zebra products, including the ET401, TC501, TC53E.



Use cases in action

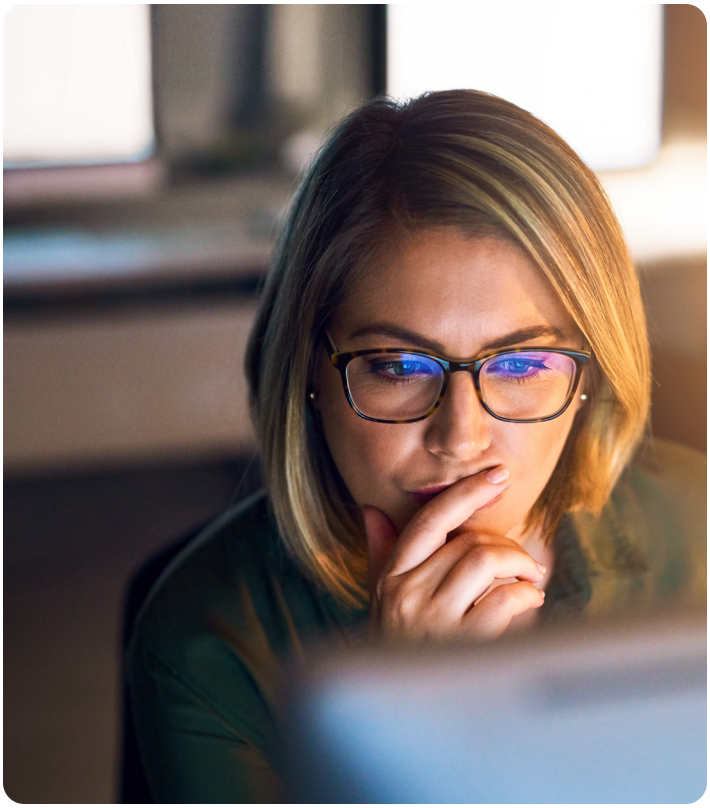
The true power of the Secure Element is realized in its practical applications. Here are a few key use cases:

- **Use case: secure validation of Apple & 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 challenge:** The business needs to validate the pass instantly and with absolute certainty that it is authentic and not a fraudulent copy.
  - **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. The private key never leaves the SE, ensuring the transaction is secure and impervious to software-based snooping. This allows 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.
  - **The challenge:** The server must verify that it is communicating with a genuine, uncompromised Zebra device.
  - **The SE solution:** The Secure Element supports key attestation. It can cryptographically prove 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.

Trusted Execution Environment (TEE)

The terms ARM TrustZone and Trusted Execution Environment (TEE) are often used together because TrustZone is the technology that makes the TEE possible on most mobile device processors.

- **ARM TrustZone:** This is a hardware feature built into ARM-based processors, the kind found in most smartphones and all Zebra mobile computers. TrustZone technology partitions the processor into two separate "worlds":
  - **Normal World (non-secure):** This is where the standard operating system, like Android, and all its applications run.
  - **Secure World (secure):** This is a hardware-isolated environment that runs a separate, smaller, and highly secure micro-kernel. It has its own dedicated memory and access to peripherals that the Normal World cannot touch directly.
- **Trusted Execution Environment (TEE):** The TEE is a secure operating environment that is created by using the Secure World of ARM TrustZone. It is where Trusted Applications (TA) run, performing sensitive tasks like cryptographic operations or processing biometric data.



In essence, TrustZone is the hardware foundation, and the TEE is the secure environment built upon it. The Android OS in the Normal World can make requests to the TEE in the Secure World, but it cannot directly access the TEE's memory or the cryptographic keys being processed within it.

Similarities and differences: TEE vs. Secure Element (SE)

Here is how a TEE (powered by TrustZone) compares to a discrete Secure Element (SE).

Key similarities

Both the TEE and the SE share the same fundamental goal: create a hardware-isolated environment for processing sensitive data and protecting cryptographic keys.

Similarity	Description
Hardware isolation	Both environments are separated from the main operating system by a hardware barrier, preventing software-based attacks (like malware on Android) from accessing the secret keys they protect.
Secure services	Both are used to provide core security services, such as key generation, secure storage of keys, and performing cryptographic operations (e.g., signing, encryption) without exposing the keys.
Protection against software attacks	They are both highly effective at defeating software-level attacks, as even a compromised operating system cannot reach into the secure environment to steal keys.

Key differences

The primary difference comes down to physical implementation and the resulting level of security, especially against hardware-based attacks.

Aspect	TEE (TrustZone)	Secure Element (SE) / StrongBox
Physical hardware	Uses a secure partition of the main CPU. It is logically separate but physically part of the same chip as the main OS processor.	A completely separate, discrete microchip with its own dedicated CPU, RAM, and secure storage.
Attack resistance	Highly resistant to software attacks. However, it is potentially more vulnerable to sophisticated hardware attacks (e.g., physical probing, side-channel analysis) since it shares the same physical processor package.	Designed from the ground up to be tamper-resistant against both software and advanced hardware attacks. The chip is purpose-built to detect and respond to physical tampering attempts.
Security certification	TEE implementations can achieve security certifications, but they typically reach a lower level than dedicated SEs.	Secure Elements often undergo rigorous, independent security evaluations and achieve high-level certifications (e.g., Common Criteria EAL5+), which certify their resistance to physical penetration.
Android terminology	When KeyMint runs in the TEE, it is referred to as a hardware-backed Keystore.	When KeyMint runs on a Secure Element, it is known as a StrongBox-backed Keystore, which Android recognizes as the highest level of security.
Primary use case	General-purpose secure operations, such as DRM (Digital Rights Management) or protecting application secrets.	Highest-security applications, such as payment processing (EMV), secure validation of transit/event passes (Apple/Google Wallet), and protecting the most critical enterprise credentials.

In summary

TEE offers a significant and robust security upgrade over software-only solutions. However, a Secure Element provides a superior level of security because it is a purpose-built, physically separate, and tamper-resistant “vault” designed to withstand a wider and more sophisticated range of attacks, including direct physical intrusion. This is why Google designates the SE-based implementation as “StrongBox” to signify this higher-grade security.

Zebra commitment to enterprise-grade security

By integrating a hardware-based Secure Element based on StrongBox, Zebra provides a platform that customers can trust with their most sensitive operations. For Zebra Sales and vendor partners, this feature is a powerful differentiator, demonstrating a clear commitment to delivering secure, reliable, and future-proof enterprise mobility solutions.

Appendix

Terminology (terms and explanation)

Term	Explanation
Secure Element (SE)	A dedicated, tamper-resistant microchip with its own processor and memory, physically isolated from the device's main processor to act as a digital vault for sensitive data like cryptographic keys.
StrongBox	A Google-specified, high-security implementation of cryptographic services that runs on a dedicated hardware security module, such as a Secure Element. It provides the highest level of security for keys on Android devices.
KeyMint	The modern software interface (HAL) in Android that allows the operating system's Keystore to access hardware-backed cryptographic functions. It acts as the standardized bridge between software and secure hardware like the TEE or a StrongBox.
KeyMint	The older, legacy version of the Hardware Abstraction Layer that KeyMint replaced. It served a similar function of connecting the Android Keystore to secure hardware.
Trusted Execution Environment (TEE)	A secure, isolated area within a device's main processor that runs separately from the main Android operating system. It provides hardware-backed security but is considered less secure than a fully separate StrongBox chip.
Hardware Abstraction Layer (HAL)	A software layer in Android that provides a standard interface for the operating system to communicate with a device's hardware components without needing to know the specifics of the hardware implementation.
Android Keystore System	A system in the Android OS that allows applications to store and use cryptographic keys in a secure manner, making them more difficult to extract from the device. It is the API that apps use to access KeyMint and StrongBox.
Cryptographic Keys	Pieces of secret data (like a password) that are used by algorithms to encrypt and decrypt data, or to create and verify digital signatures. Protecting these keys is fundamental to security.
Key Attestation	A process where the Secure Element or TEE cryptographically proves that a key it is using is genuinely protected by secure hardware. This allows a remote server to verify that it is communicating with a legitimate, uncompromised device.
Apple VAS / Google Wallet	Digital wallet platforms from Apple and Google that allow users to store items like event passes, loyalty cards, and tickets on their mobile devices for contactless use.
Trusted Application (TA) / Applet	A small, secure application that runs inside the Trusted Execution Environment (TEE) or on the Secure Element (SE). It handles sensitive operations, ensuring they are isolated from the main OS.
Open Mobile API (OMA)	A standard API that defines how applications on a device's main processor can communicate with the applications (applets) running inside the Secure Element.
Rollback Resistance	A hardware-enforced security feature that prevents an attacker from restoring a device to a previous, less secure software version or from restoring a deleted key.
Replay Protected Memory Blocks (RPMB)	A type of secure storage that can only be written to in a way that prevents an attacker from "replaying" old data to trick the system. It is often used to implement rollback resistance.
Elliptic Curve Diffie-Hellman (ECDH)	An industry-standard cryptographic protocol that allows two parties to securely establish a shared secret over an insecure channel, used for securely exchanging keys.
ARM TrustZone	A hardware security extension built into ARM processors that partition the CPU into two isolated "worlds": a non-secure Normal World for the main OS and a Secure World where a TEE can operate.

→ **For further information please visit www.zebra.com/security**

ZEBRA