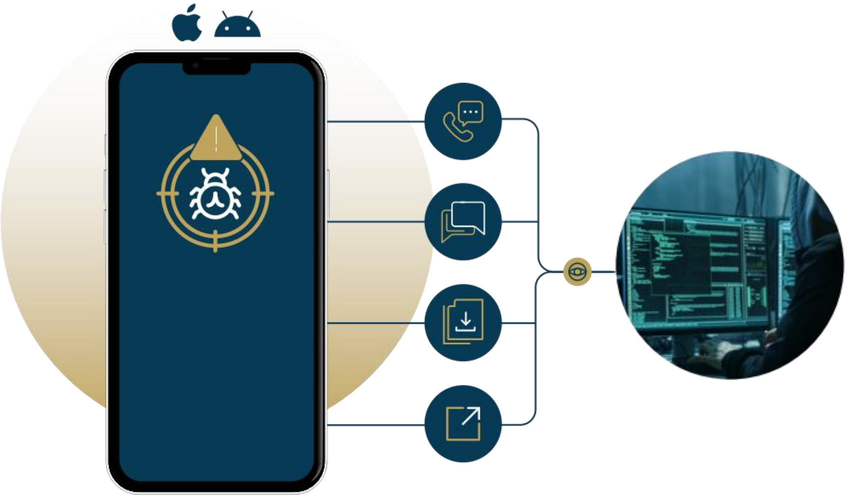
THE WORLD HAS CHANGED

The need for total mobile privacy and security has never been greater. Common mobile phone operating systems iOS and Android are not capable of shielding phones from spyware such as Pegasus.



Hackers exploit the vulnerabilities in commercial operating systems, particularly Android or iOS, and messaging applications such as Signal or WhatsApp.

For a mobile phone to be completely secure, it must protect the applications, the operating system, and the hardware. Only a comprehensive approach that addresses every aspect of your mobile communications can ensure you have absolute protection.

INTRODUCING THE SOTERA SECUREPHONE

Trusted globally by governments, enterprises, and individuals as their safe place for voice and text communications.

The Sotera SecurePhone uses the same operating system (Integrity 178B) that secures the United States nuclear arsenal, multiple weapon systems, and NASA/DOD space systems.

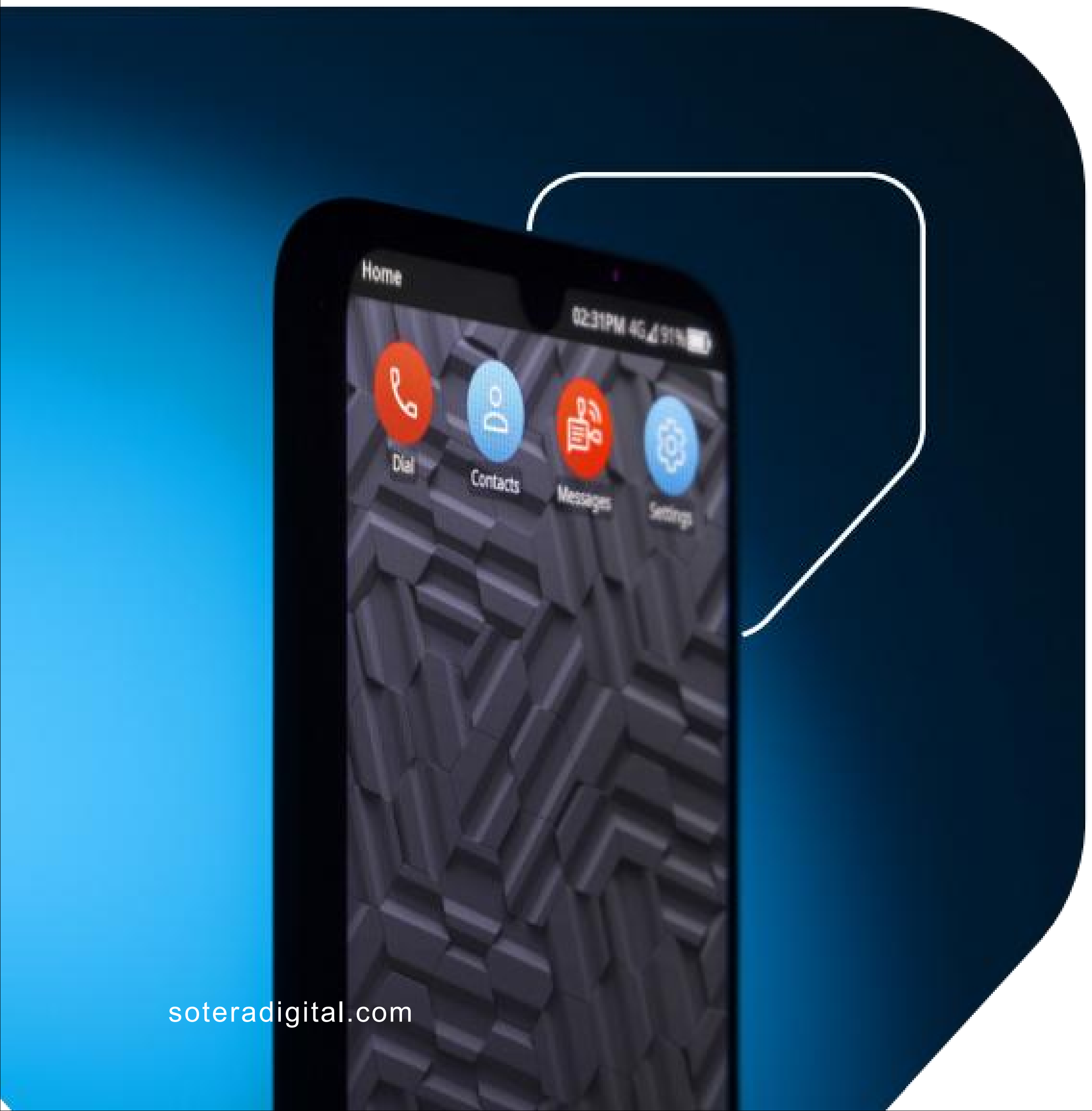
Only a multi-layered software and hardware security solution can successfully protect your data.

Layer 1: The Applications.
Encrypted VoIP apps like WhatsApp, Signal and Telegram don't protect your communications end-to-end.

Layer 2: The Operating System.
Apple and Android operating systems are not secure and have thousands of known vulnerabilities.

Layer 3: The Hardware.
Standard smartphones do not protect against malware threats on either endpoint device.

SOTERA ENCRYPTION



Disk Encrytion:

XTS-AES256

Secure text and messaging exchange and encryption:

We use the Double Ratchet protocol to exchange encrypted messages. X3DH is used for key agreement between parties.

Under the hood, the algorithms used by these protocols are:

EdSA with Curve225519 (Ed25519)
ECHD with Curve 25519 (X25519)
AEAD encryption scheme using HKDF with AES256-CBC / HMAC-SHA256

Voice Exchange and Encryption:

Shared secret exchanged thru the secure messaging channel between devices:
Voice encrypted using AEAD AES256-GCM after deriving using HKDF an AES256-CM_PRF from shared secret.

Additionally for nettwork traffic we layer on:

TLS 1.3 using ED25519/X25519 and mutual authentication.

XEdDSA and VXEdDSA: Used to create and verify EdDSA-compatible signatures using public key and private key formats initially defined for the X25519 and elliptic curve Diffie-Hellman functions. It also uses 'VXEdDSA,' WHICH EXTENDS XEdDSA to make it a verifiable random function, or VRF.

X3DH: 'Extended Triple Diffie-Hellman' is a key agreement protocol. X3DH establishes a shared secret key between two parties who mutually authenticate each other based on public keys. X3DH provides forward secrecy and cryptographic deniability.

Double Ratchet: The Double Ratchet algorithm, which is used by two parties to exchange encrypted messages based on a shared secret key. The parties derive new keys for every Double Ratchet message so that earlier keys cannot be calculated from later ones. The parties also send Diffie-Hellman public values attached to their message. The result of Diffie-Hellman calculations is mixed into the derived key so that later keys cannot be calculated from earlier ones. These properties give some protection to earlier or later encrypted messages in case of a compromise of a party's keys.

Sesame: The Sesame algorithm is for managing message encryption sessions in an asynchronous and multi-device setting.