Mobile Application Protection

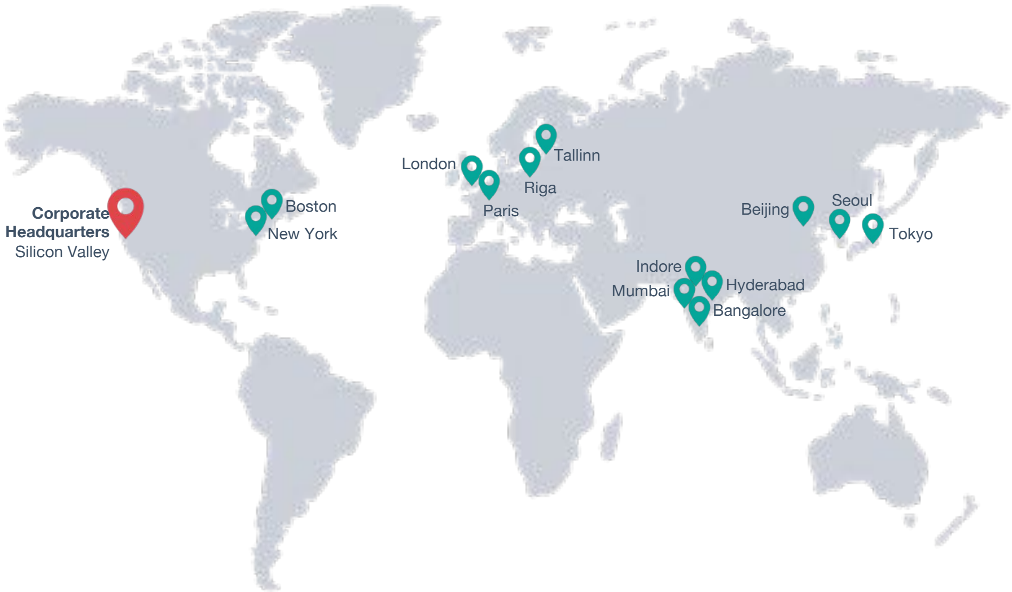
Bill Horne, VP and GM
Intertrust Secure Systems

intertrust

November 9, 2017

intertrust

- Over 25 years of experience in security and trusted computing
- Headquartered in Silicon Valley with global offices in North America, Europe and Asia
- Leading content protection and rights management technology provider
- Global customer base in consumer electronics, mobile, automotive, healthcare and enterprise with billions of devices protected
- Extensive intellectual property portfolio
- Privately held. Investor base includes Sony, Philips, WiL and innogy SE



intertrust

CONTENT MANAGEMENT

ExpressPlay™

DRM System

Cloud-based content distribution system for video, audio, and eBooks.

Kiora™

Offline Content Delivery System

Secure content distribution platform for low-bandwdith domains.

SECURE SYSTEMS

whiteCryption™

Application Shielding

Tools to prevent reverse engineering and tampering.

Seacert™

Certificate Authority

Large scale cryptographic key provisioning and managed PKI.

TRUSTED DATA PLATFORM AND SERVICES

Personagraph™

Customer Data Platform

Custom targeted segments derived from first-party app data, CRM databases, and offline purchases, ensuring that advertisers can always identify and reach their most valuable customers.

Planet OS™

Geospatial Big Data Platform

Big data infrastructure to help renewable energy companies transform the way data is used in their organizations.

Genecloud™

Genomic Data Platform

Trusted cloud service for storing and analyzing genetic sequence data, balancing access and privacy.

Computing is Evolving

Mobile Devices & Cloud Services

IoT & Embedded Systems

Implications

Profound impact on user experience and the way we do business

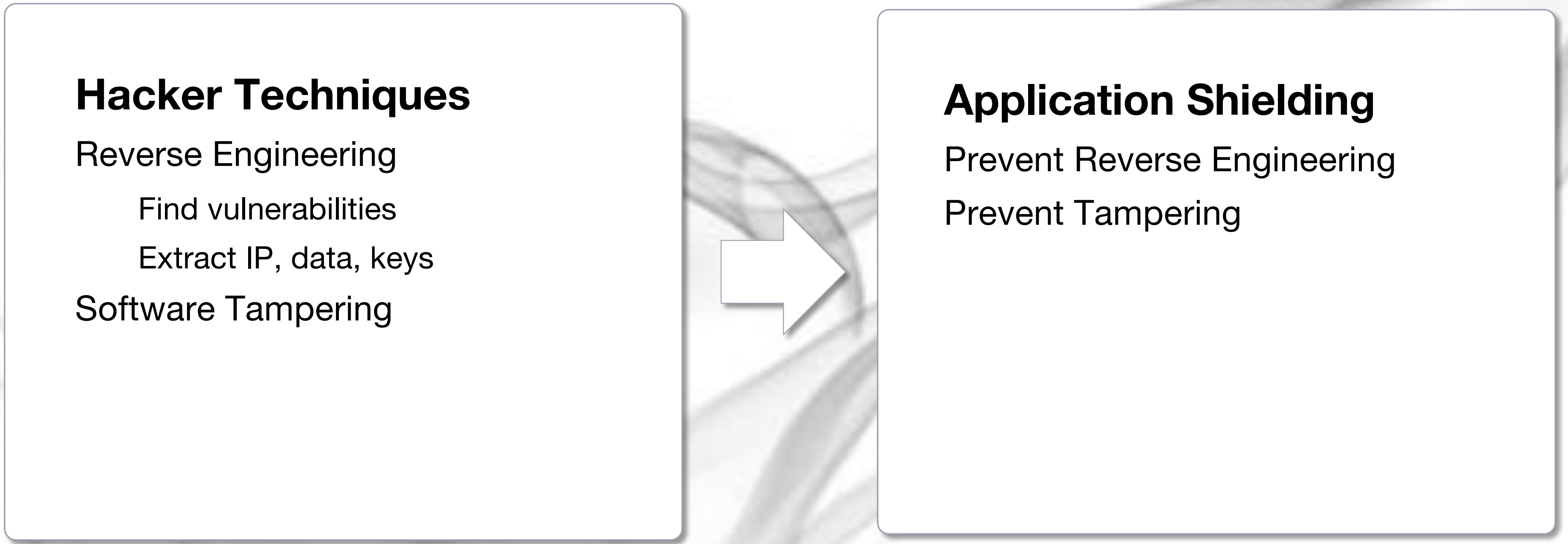
Hackers have unprecedented physical access

Hacker Goals

- Bypass Business Logic
- Steal Intellectual Property
- Steal Sensitive Data
- Obtain Cryptographic Keys
  - Steal Content
  - Masquerade as Users/Devices
  - Snoop on Communications
- Stepping Stone Attacks

Consequences

- Financial Loss
- Brand Reputation
- Liability



Hacker Techniques

Reverse Engineering

- Find vulnerabilities
- Extract IP, data, keys

Software Tampering

Application Shielding

Prevent Reverse Engineering

Prevent Tampering

How do you prevent reverse engineering and tampering?

Key Idea #1: Code Obfuscation

Goal

- Make it as difficult as possible to understand what software is doing

Techniques

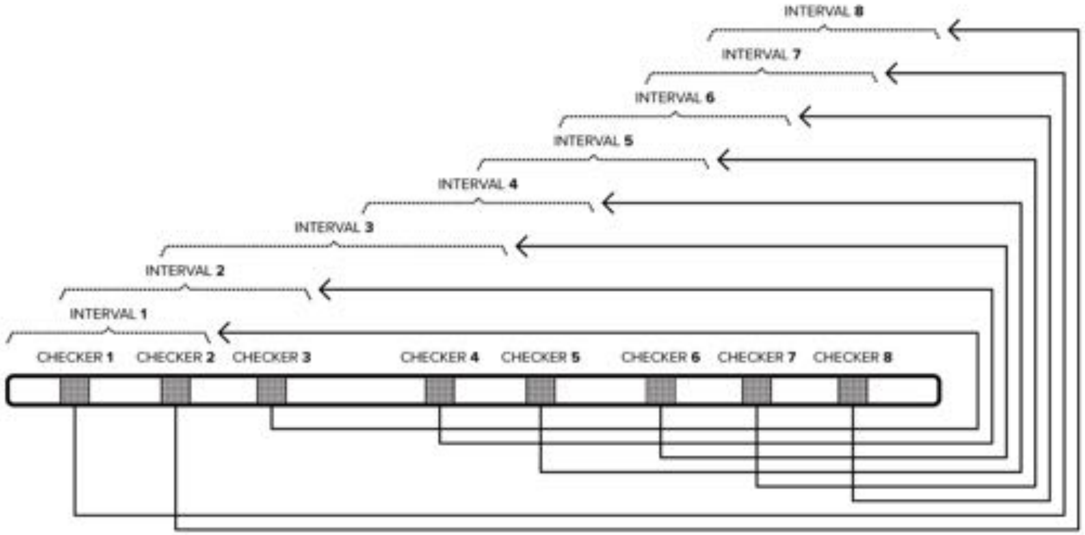
- Symbol Stripping and Renaming
- String Renaming & Encryption
- Opaque Predicates
- Basic Block Splitting and Merging
- Control Flow Obfuscation
- Code Flattening
- Function In-lining

```
static DATA *z16208f39bf; static char*zb37c9d1346 = ""; static
int z388d3293ac = (0xba4 + 1467-0x115e) ; static int
z99ec214447; static int za862d19cbc; static int z1c0ab7cf0c
; static int z13f00839ad =-(0x10a5 + 1725-0x1761) ; static int
z22204afdf5; static long zbbec3834b1; static void z6aea0a920d
(char*s) { perror (s) ; exit (EXIT_FAILURE) ; } static void
zfe178f875a (int zab628eb42a) { if (z99ec214447) { (void) fputc
(zab628eb42a, stderr) ; (void) fflush (stderr) ; } } static
char*z66f17a4c78 (char*s) { return strcpy (malloc ((unsigned)
(strlen (s) + (0xffa + 212-0x10cd))) , s) ; } static DATA*
z4f6e1f2cad (char*name) { register DATA*z04eb77b88a,
*z34d15a68ff, *z04526a1d1b; z7f02667ab7 ((
"\x6e\x65\x77\x5f\x64\x61\x74\x61\x28\x25\x73\x29" "\n", name))
for (z04eb77b88a = z16208f39bf, z34d15a68ff = (0x49b + 7318-
0x2131) ; z04eb77b88a! = (0x5d0 + 3794-0x14a2) ; z34d15a68ff
= z04eb77b88a, z04eb77b88a = z04eb77b88a ->link) { int
z327a26f629 = strcmp (z04eb77b88a ->name, name) ; if
(z327a26f629 == (0xe0 + 6557-0x1a7d)) return z04eb77b88a; if
(z327a26f629> (0xe20 + 1631-0x147f)) { break; } } z04526a1d1b =
(DATA *) malloc (sizeof (DATA)) ; if (z34d15a68ff! = (0x166
+ 2883-0xca9)) z34d15a68ff ->link = z04526a1d1b; else
z16208f39bf = z04526a1d1b; z04526a1d1b ->link = z04eb77b88a
; z04526a1d1b ->name = z66f17a4c78 (name) ; z04526a1d1b ->base
= (0x4ac + 6313-0x1d55) ; z04526a1d1b ->z25cd54603c
= ze04ece0484; z04526a1d1b ->zc00bf817a3 = z04526a1d1b ->
z9cbe16f057 = z04526a1d1b ->z8d69073f9f = (0xe13 + 195-0xed6)
; return z04526a1d1b; }
```

Key Idea #2: Integrity Checking

Goal

- Continually check the code to make sure that it hasn't been modified
- If it has been modified, take an appropriate action
- Runtime Application Self Protection (RASP)
- Published and patented



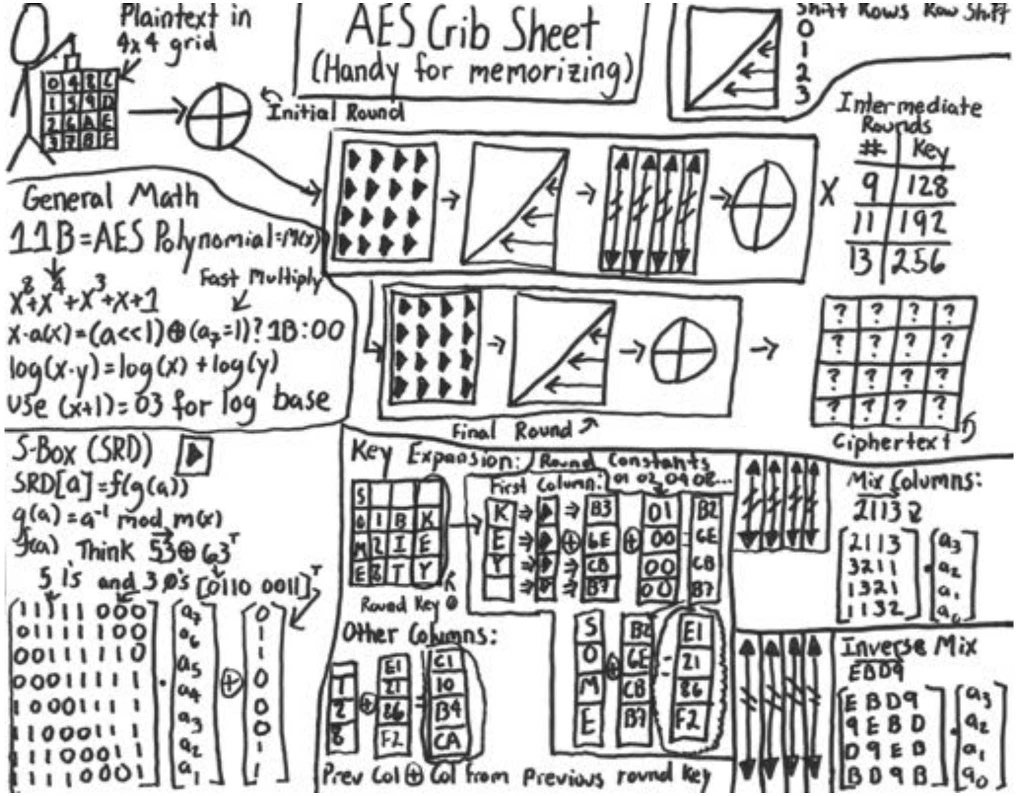
Key Idea #3: White Box Cryptography

Goal

- Implement standard cryptographic operations without the key ever being in the clear

Properties

- Static, dynamic and wrapped keys
- Resistant to side channel attacks
- Support a wide variety of cryptographic algorithms



Additional Techniques

Anti Reverse Engineering

- Debugger Detection
- Binary Packing
- Diversification

Anti Tampering

- Anti-method Swizzling
- iOS Jailbreak Detection
- Android Rooting Detection
- Function Caller Verification
- Shared Library Cross-Checking
- Mach-O Binary Signature Verification
- Google Play Licensing Protection

whiteCryption
Code Protection

Provides mobile apps and IoT devices with code obfuscation and Runtime Application Self Protection (RASP), shielding them from decompilers, debuggers, reverse engineering and tampering by hackers.

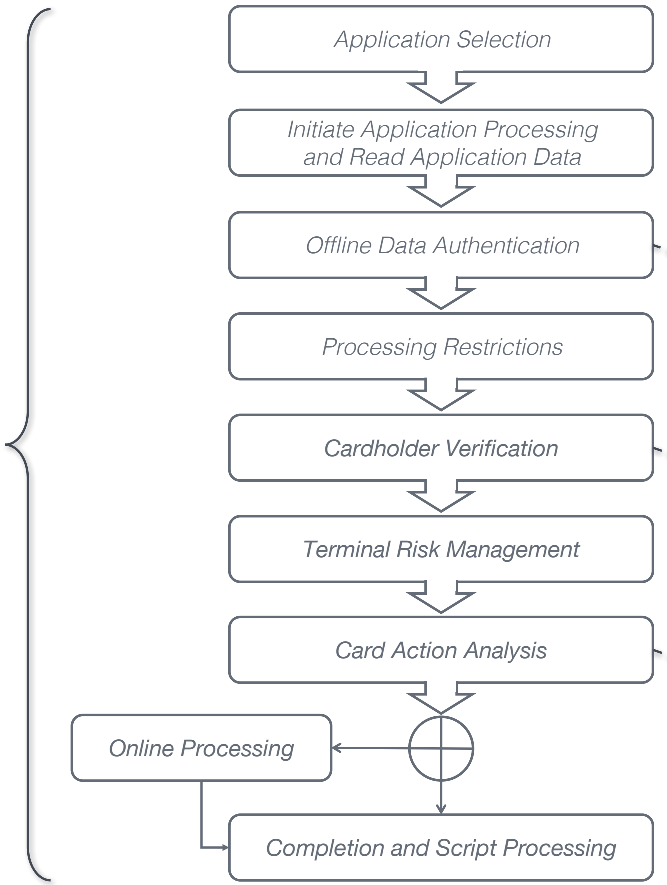
whiteCryption
Secure Key Box

Provides mobile apps and IoT devices with a drop-in replacement cryptographic library that performs common cryptographic functions while keeping secrets and cryptographic keys secure.

SOFTWARE BASED HOST CARD EMULATION FOR PAYMENTS

whiteCryption
Code Protection

Protect the entire app from reverse engineering and tampering



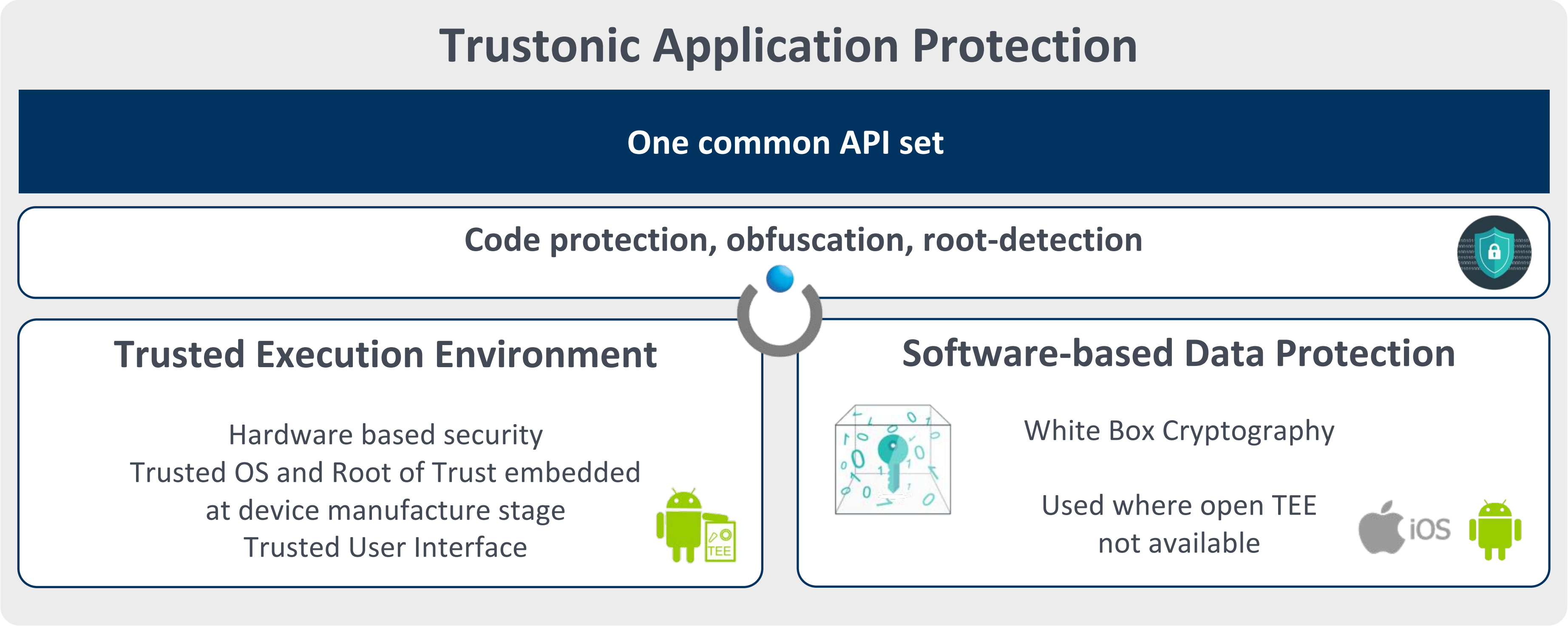
whiteCryption
Secure Key Box

Static, Dynamic, or Combined Data Authentication (SDA, DDA, CDA)

“Enciphered” PIN incorporated into online mode as well as one offline mode

Card signs transaction information to be sent to issuer, issuer responds with signed data

Optimum Protection Across Fragmented Devices



Trustonic Application Protection

One common API set

Code protection, obfuscation, root-detection

Trusted Execution Environment

Hardware based security
Trusted OS and Root of Trust embedded
at device manufacture stage
Trusted User Interface

Software-based Data Protection

White Box Cryptography

Used where open TEE
not available

THANK YOU

intertrust®

www.intertrust.com

SKB SUPPORTED CIPHERS AND ALGORITHMS

Encryption
AES-128/192/256 (ECB, CBC, CTR) , DES & 3DES (ECB and CBC)

Decryption
AES-128/192/256 (ECB, CBC, CTR) , DES & 3DES (ECB and CBC) , RSA-1024/2048 (OAEP or v1.5) , El Gamal Elliptic Curve Cryptography (ECC)

Authenticated Encryption
AES-128/192/256 (GCM)

Signing
AES-CMAC, HMAC, RSA Signature, RSA Probabilistic Signature, ECDSA.

Verification
AES-CMAC, HMAC, ISO/IEC 9797-1 MAC (Retail MAC)

Authentication
CDMA2000 authentication algorithm

Key Generation
Random buffer of bytes for AES, DES, 3DES algorithms; key pairs for Elliptic Curve Cryptography algorithms

Key Agreement
Classic Diffie-Hellman (DH) , Elliptic Curve Diffie-Hellman (ECDH)

Calculate Digests
MD5, SHA-1/224/256/384/512

Key Derivation
Large variety of key manipulation routines iterated SHA-1, SHA-256, SHA-384, byte reversing, NIST 800-108 key derivation, Open Mobile Alliance KDF2, CMLA key derivation, AES key derivation, and more.