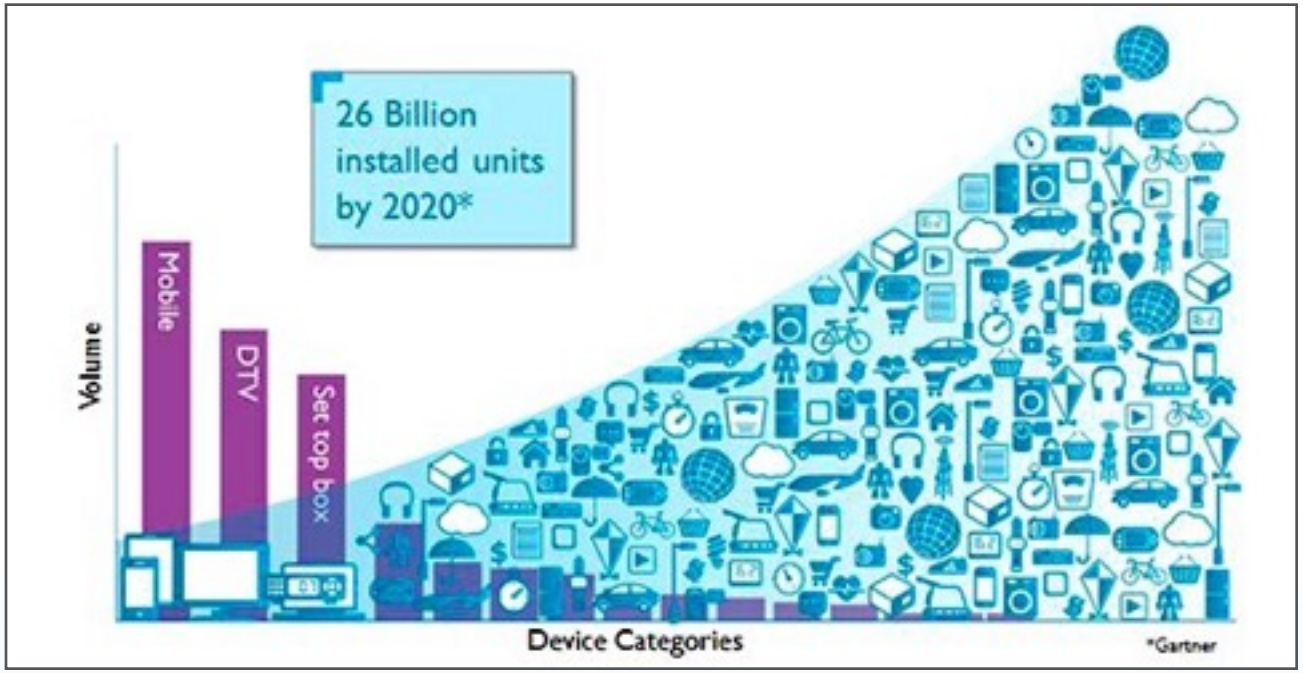
whiteCryption®

给网路服务和应用提供可信计算环境

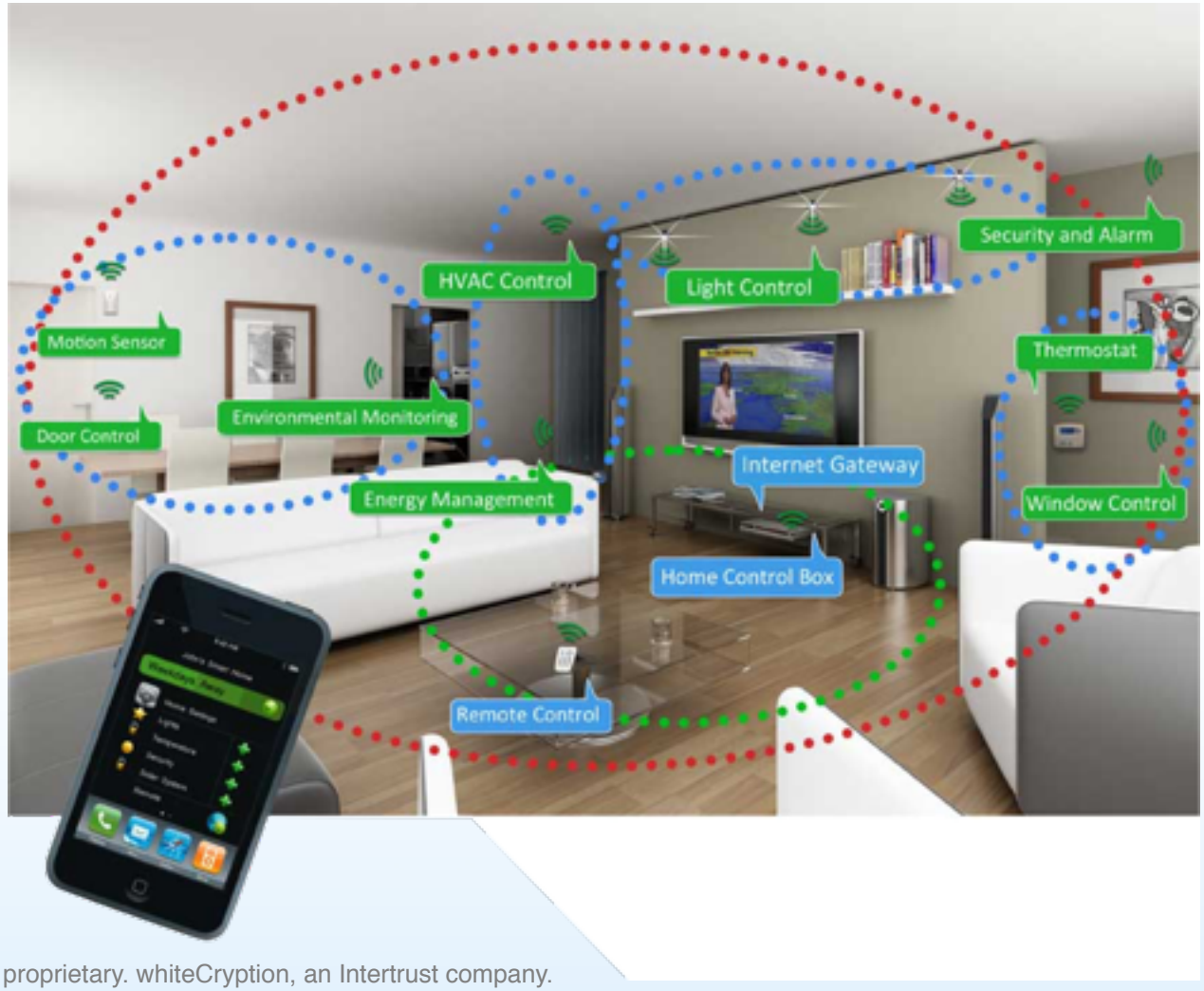
Kenny Huang
Intertrust, 2016-1-22

我们生活在万物互联的时代



智能家居 –
谁能操控你的屋子?

whiteCryption®



互联网摄像头 – 谁在偷看你的宝宝?

智能汽车 –
谁拿了你的车钥匙?

whiteCryption®

自我驾驶 – 谁在开你的车子?

移动支付 –
谁在用你的信用卡?

whiteCryption®

数字医疗 –
谁拥有我的个人隐私数据?

信任 –
智能产品的重要的品牌元素

Can someone hack my house?
Can someone hack into my car and steal it?
Can someone steal my credit card info?
Is my personal information secure?

When people trust a brand…

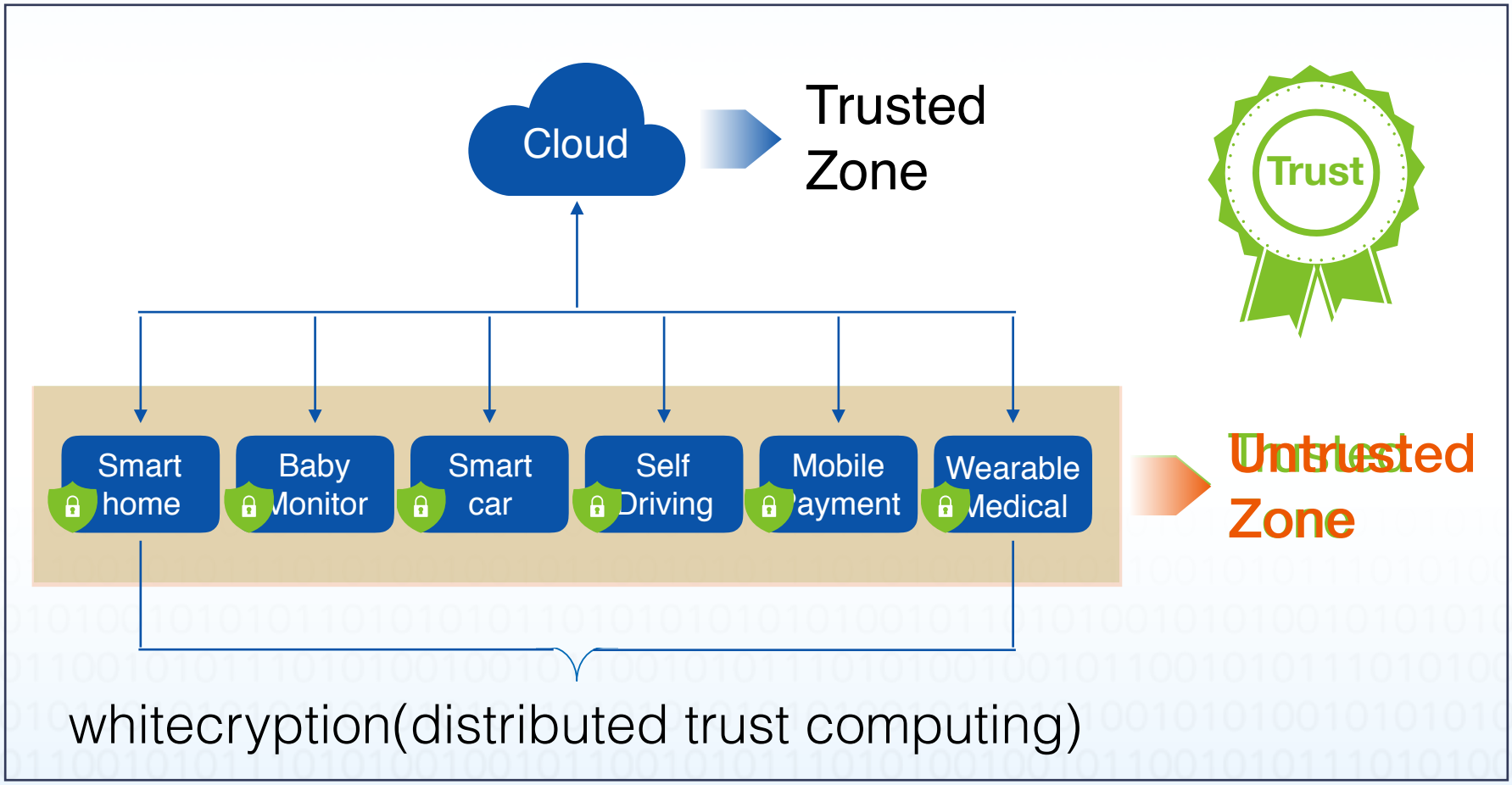
83% will **recommend** it to **other people**

82% will **use** its products & services **frequently**

50% will **pay more** for its products & services

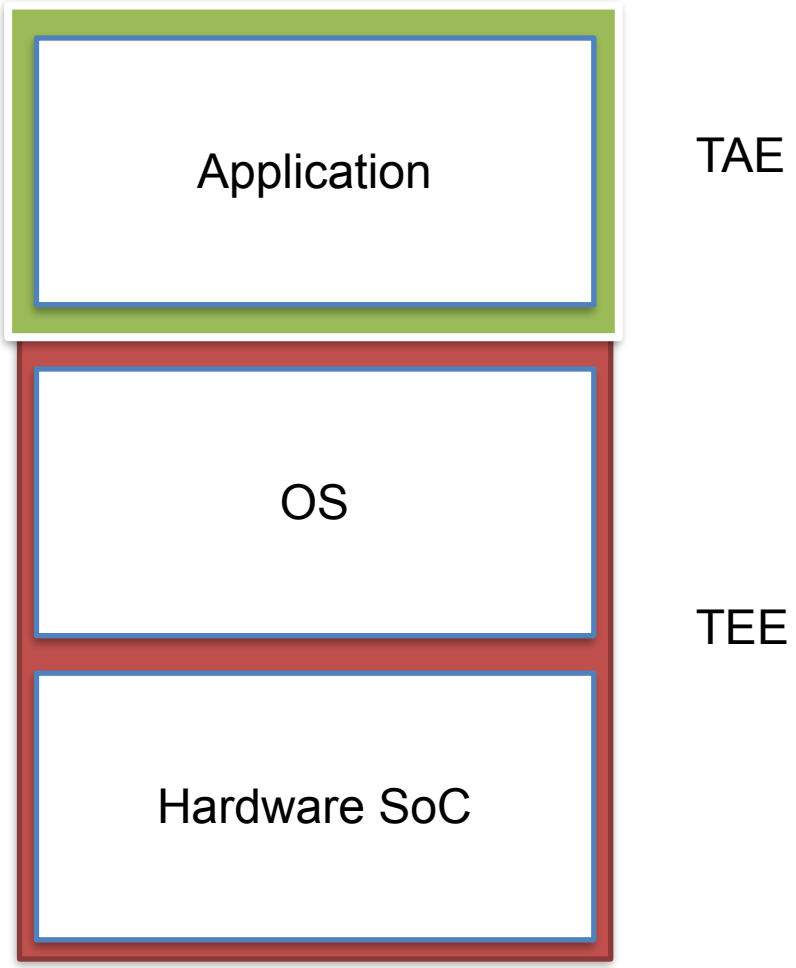
Source: Brand Trust Study (2009), Concerto Marketing Group

可信任应用环境
Trusted Application Environment (TAE)



建构可信任计算环境

- 在没有TEE设备下保障APP的执行是安全可信的
- 在有TEE的设备环境调用TEE接口, 使用TEE功能
- TAE 和 TEE 互补提供一个完整的可信计算环境



可信任应用环境 – 保护应用代码和数据

Secure Key Box

Cryptographic library that implements standard cryptographic algorithms in a way that completely hides the keys

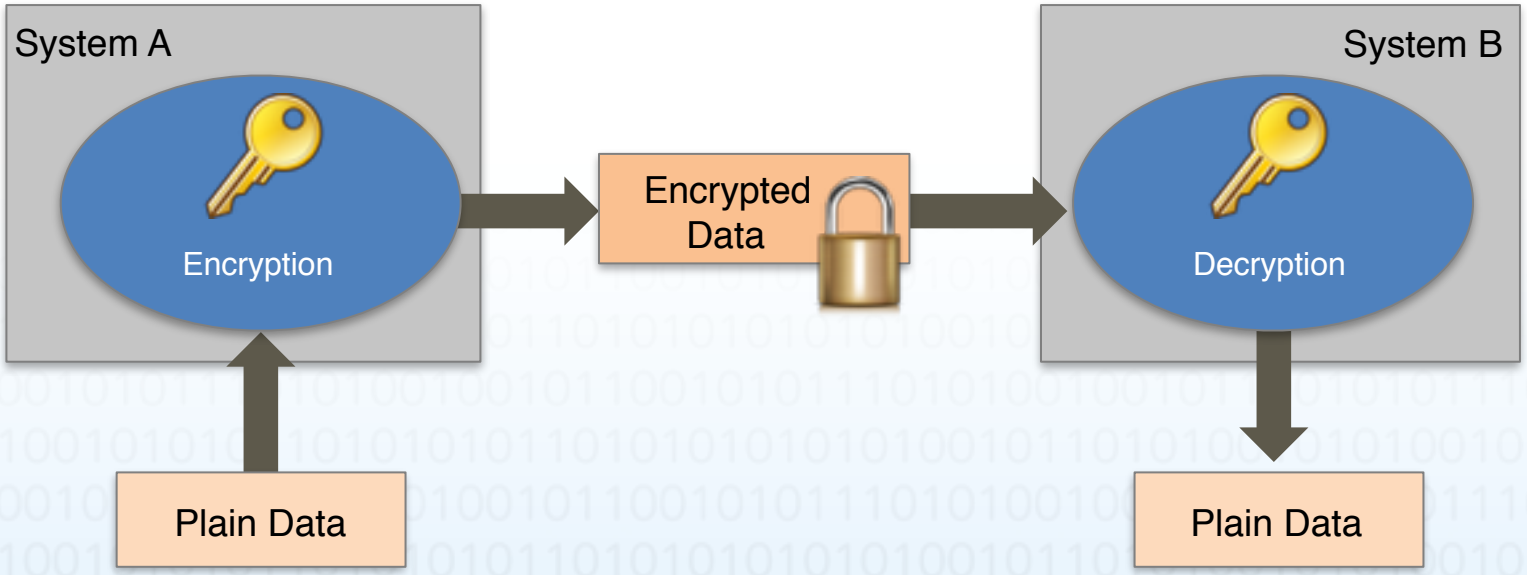
Code Protection

Comprehensive tool for hardening modern software applications on multiple platforms

SKB – 白盒密码

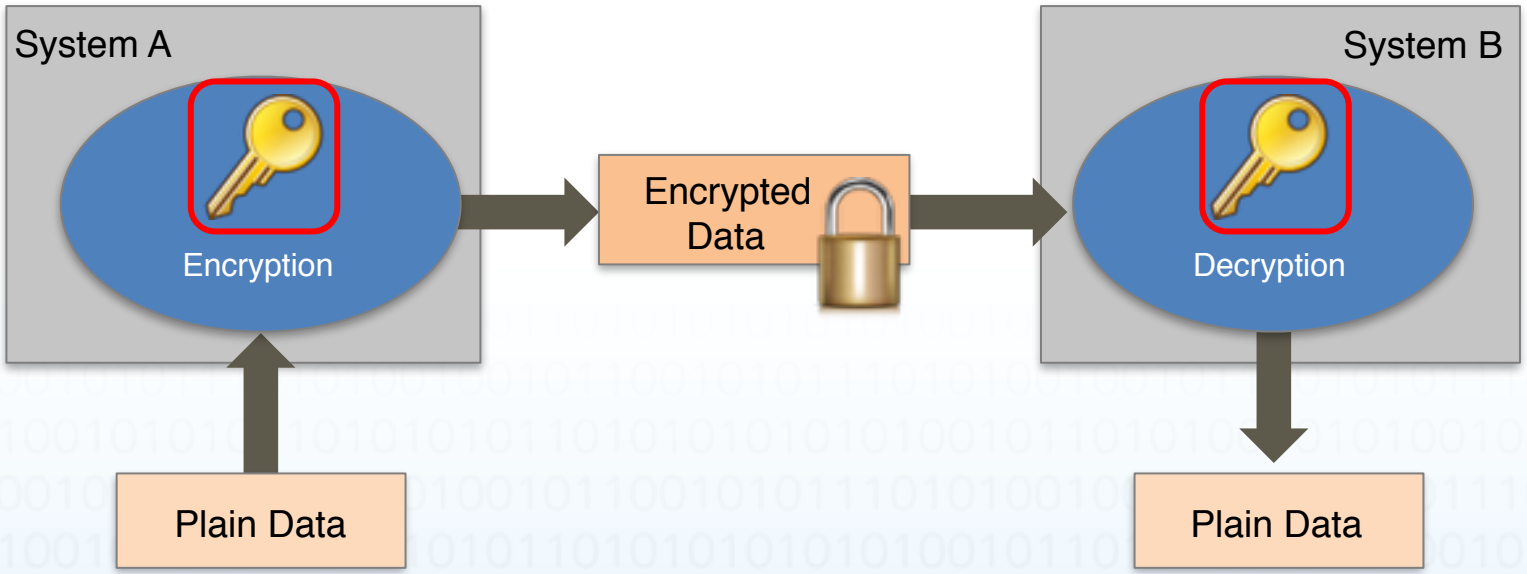
一般保护数据的方法是加密。主要加密算法比方AES，RSA，ECC等都是我们常用的算法。

例子: 为了确保两个系统中的通讯是安全的，将通信数据加密



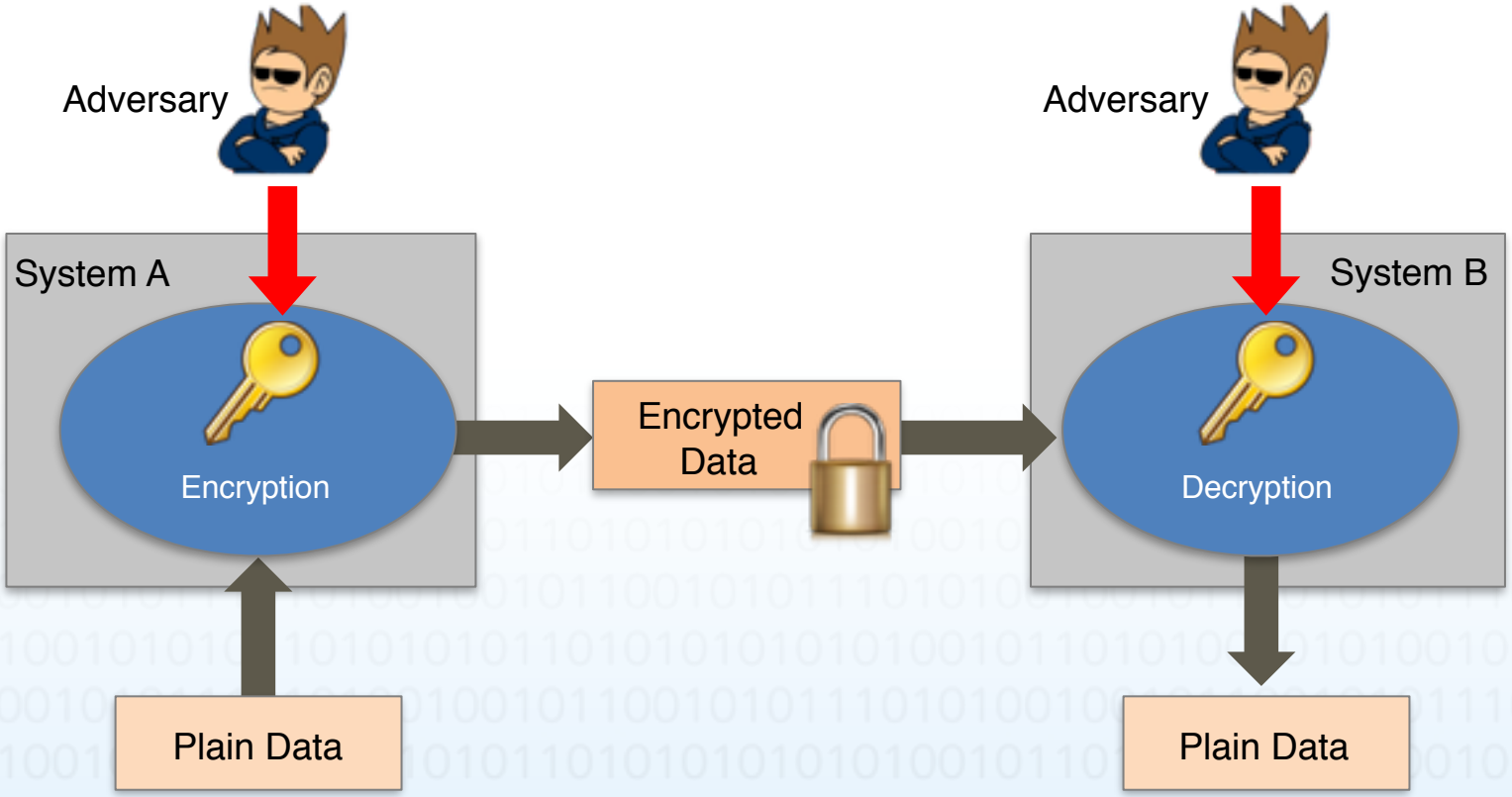
SKB – 白盒密码

问题: 加密和解密的软件非常容易把所使用的密钥曝露出来



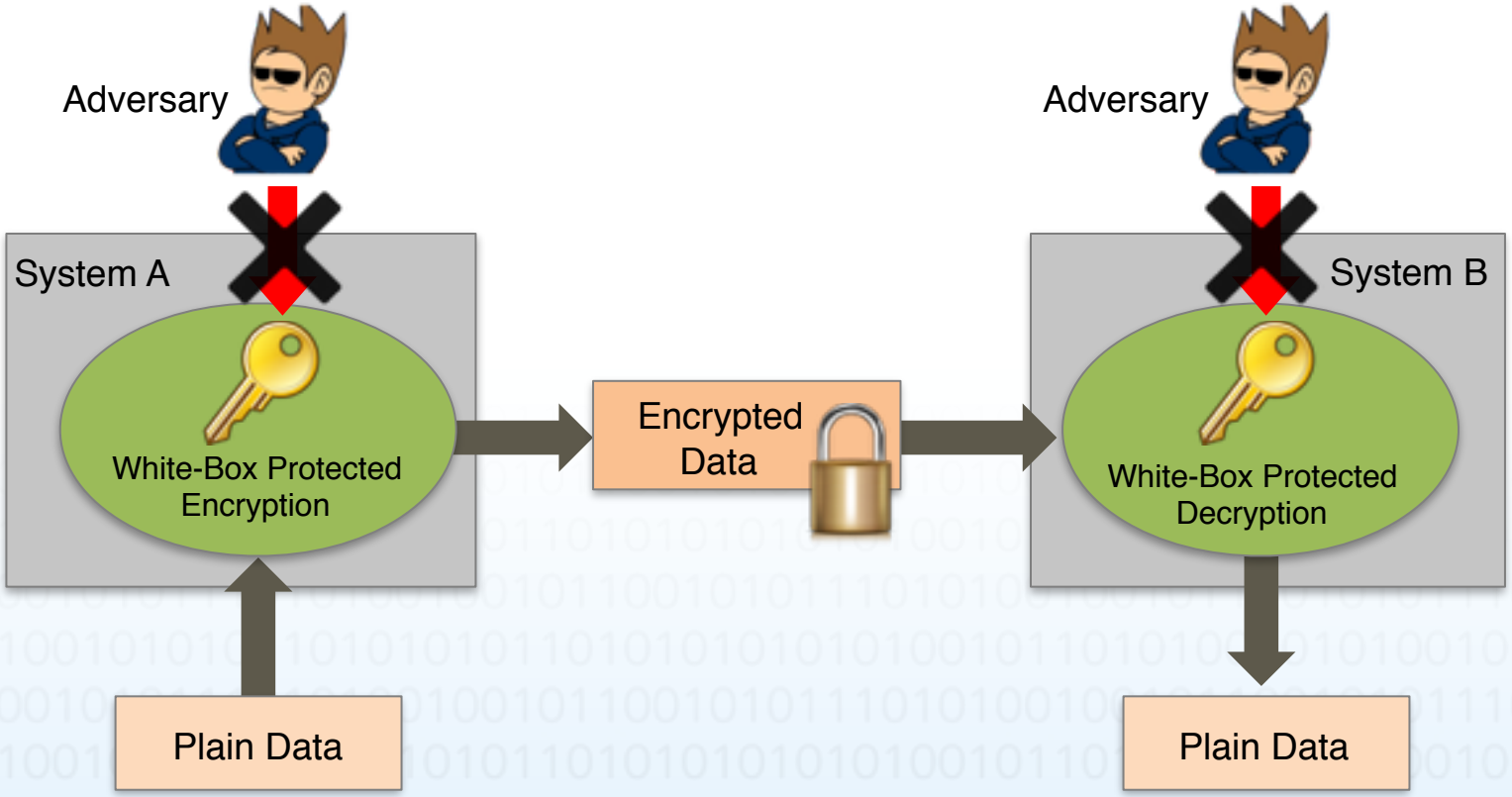
SKB – 白盒密码

外部攻击只要掌握了加密所用的密钥就可解开所保护的数据



SKB – 白盒密码

白盒密码保护加解密算法免于反编译和逆向工程的攻击



SKB – 白盒密码

白盒密码主要用来保护秘钥在不可信的环境中当执行加密解密算法时不会被泄漏

Desktop Computers

Set-Top Boxes

Embedded Systems

Mobile Devices

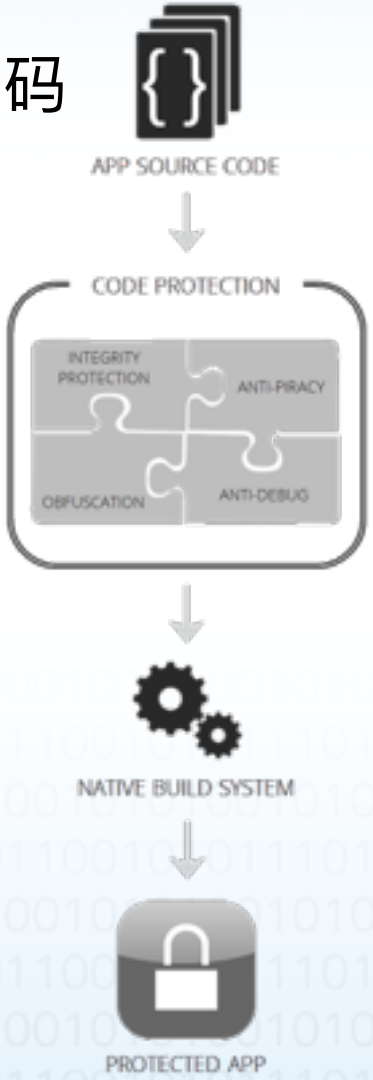
SKB – 白盒密码

whiteCryption's 白盒技术提供

- 基于白盒的以下算法
  - RSA and ECC
  - AES
  - SHA
- 如果设备有基于硬件的Crypto, SKB也可以直接调用硬件Crypto

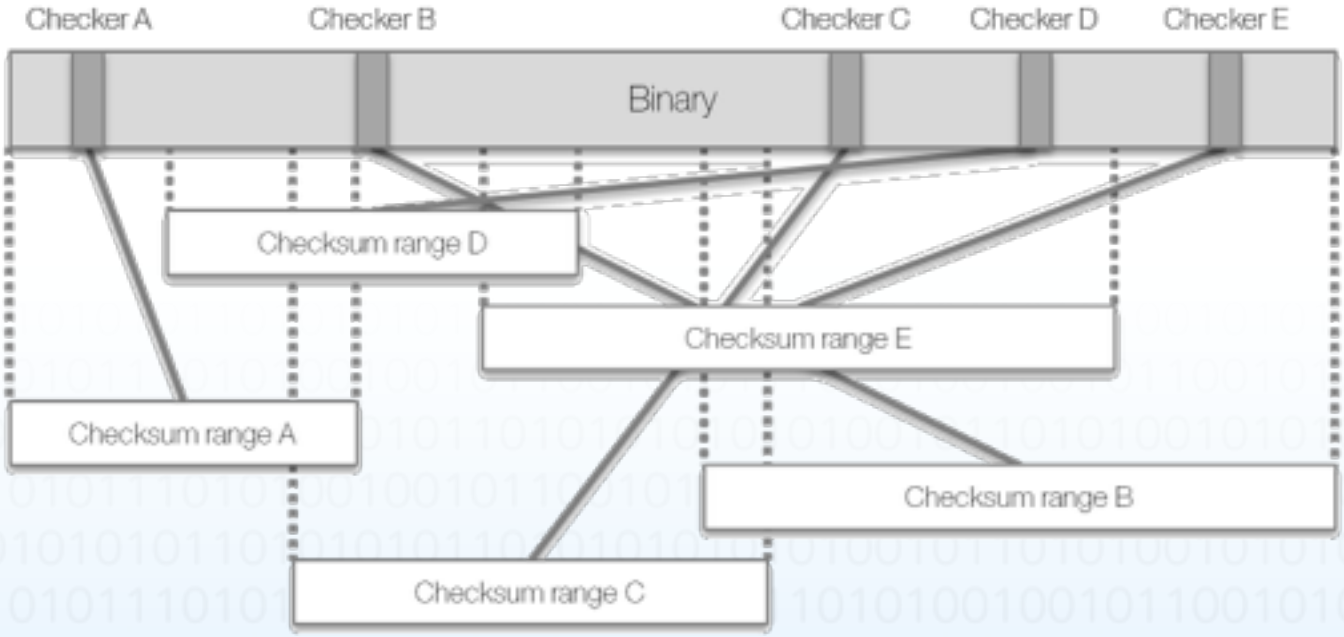
代码保护

- 保护标准 C/C++/Objective-C and JAVA 源代码
- 不需改变原始代码和编译链
- 简单而且可以自动化
- 不需改变目前你的开发环境
- 完全基于软件
- 支持自动优化
- 支持跨平台
- 支持代码多元化



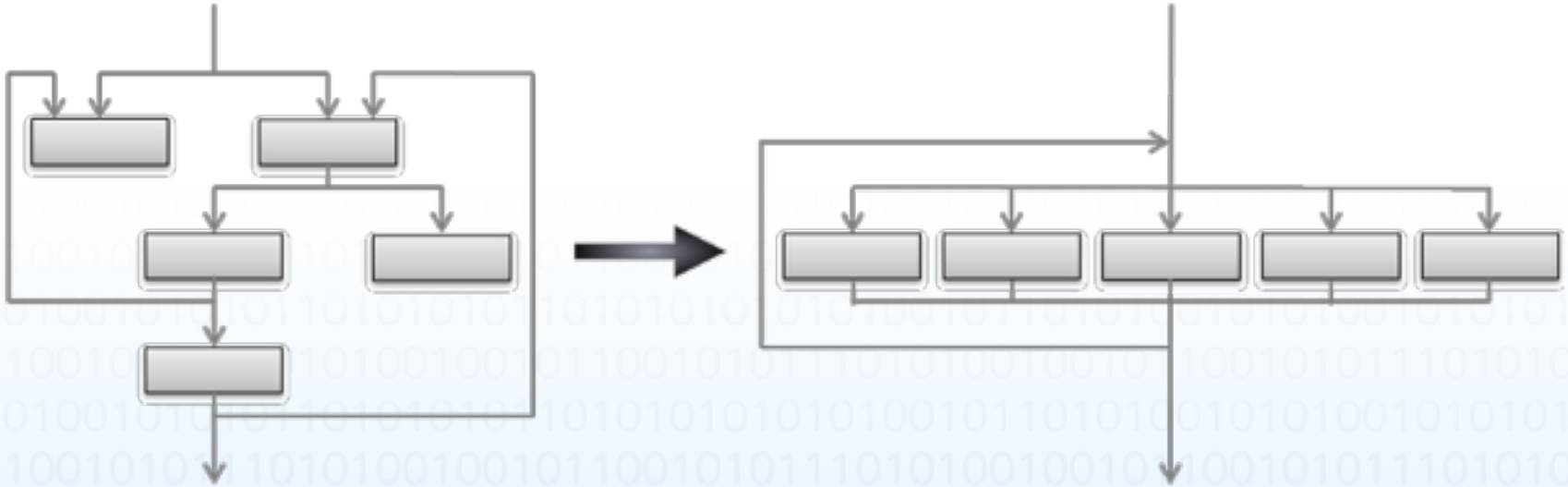
完整性保护

- 在代码中注入多个完整性检测。
- 如果发觉被保的code 被人篡改，或有code lifting的时候
- 可以自动设定crash, 或停止继续执行，或限制其执行的范围。



代码混淆

- 专利技术通过控制流扁平化算法混淆代码
- 使代码分析不可能达成
- 代码中所使用的 String literal 乱码混淆
- Objective-C message call 乱码混淆



盗版保护

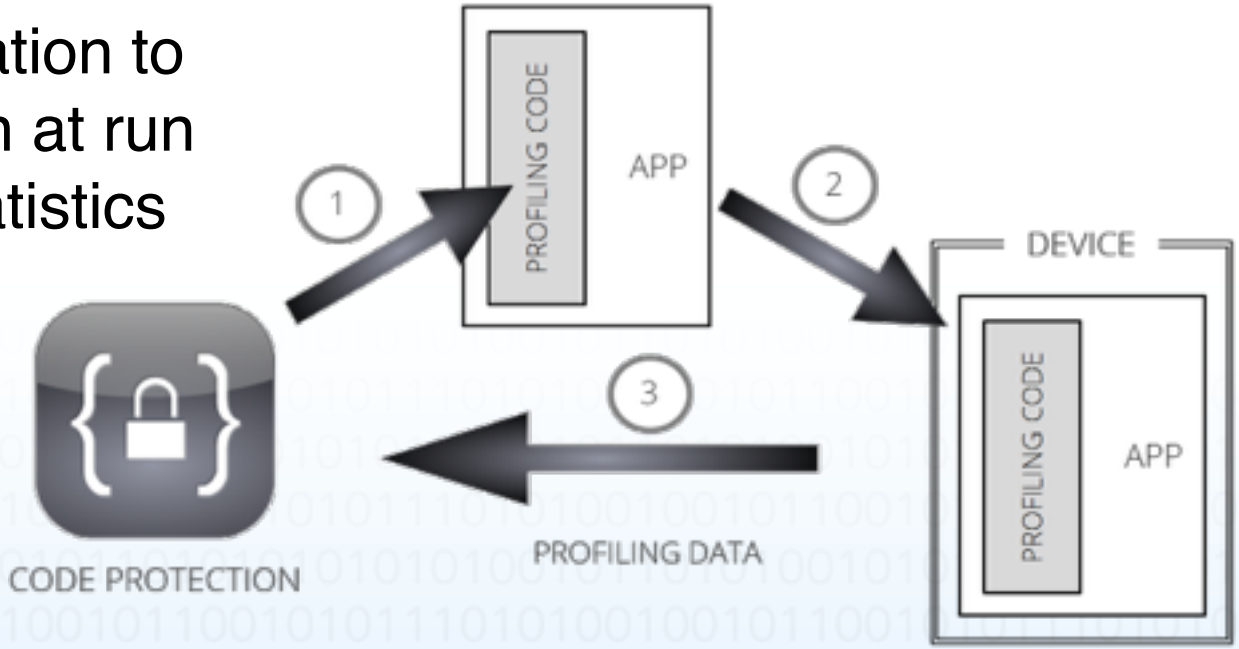
- 越狱/rooting 侦测 (iOS, Android)
  - Protected app terminates execution if the device is jailbroken/rooted
- Method 篡改侦测 (iOS)
  - 侦测被保护的APPMethod被篡改数字二次签名验证 (OS X and iOS)
- 防止APP被二次签名后再分发
  - Google Play 证书保护 (Android)
  - 使用 native-code implementation of LVL instead of the standard Java implementation
- APK packages 完整性保护 (Android)
  - 受保护的APK或签名如果被篡改则不能被继续执行

其它保护功能

- Anti-debug
  - Numerous instances of platform-specific anti-debug code are inserted into the code
- Binary packing
  - Executable code is decrypted only at run time
  - Makes static analysis much more difficult
- Function caller module verification
  - Protected functions check where they are called from
- Cross-checking of shared libraries
  - Ensures shared libraries are not modified or replaced
- Customizable defense action
  - Either crash the application or write a custom callback function to be invoked when a threat is detected

Profiling

1. Code Protection embeds profiling code into the original application
2. Developer installs and runs the application on a device
3. Application sends profiling information to Code Protection at run time or via a statistics file



多样化

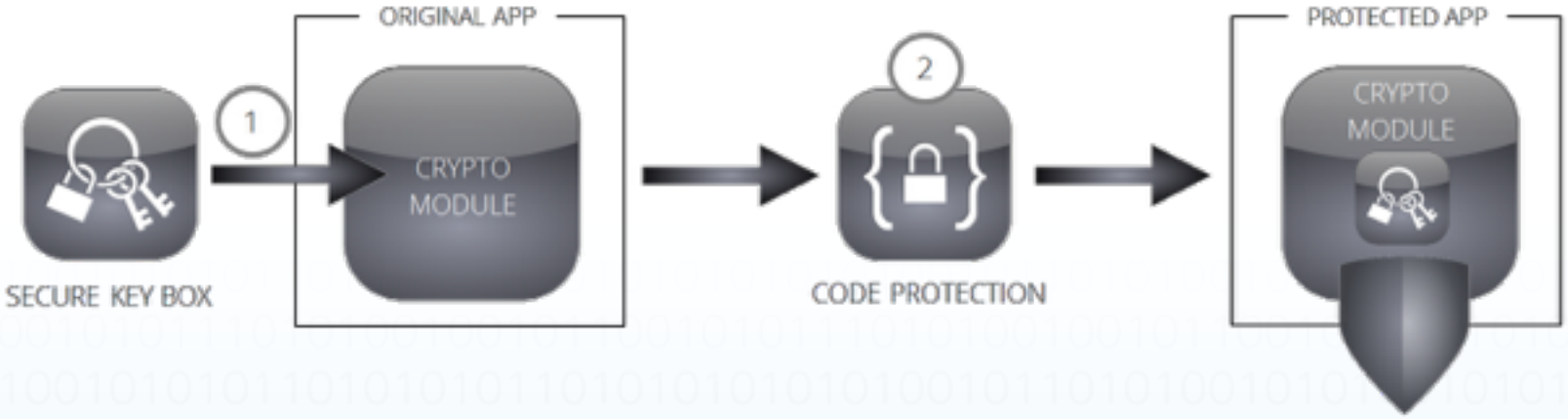
- 代码多样化
  - 经过Code Protection保护之后代码可以产生各种不同功能相同的binary
  - 可以同时产生很多不同的 SKB给同一个APP使用
- 数据多样化
  - 每个SKB所存储的密钥的格式都不同
- 优势
  - 如果一个系统被攻破，其它系统不会被影响
  - 很难产生全面的universal cracking tools
  - 保护不同系统内的数据不会被分享(数据隐私)
  - 通过快速软件更新可以防止应用攻破的影响

支持环境

- 系统平台
  - Windows
  - OS X
  - Linux
  - Android
  - iOS
  - Windows Phone (SKB only)
- 代码语言
  - C/C++
  - Objective-C
  - Java

如何有效达到可信应用环境(TAE)

- 将敏感的密码算法改为用SKB
- 使用 Code Protection features 来保护整个 code base



谢谢

给网路服务和应用提供可信计算环境

Become a Partner

Contact:
Kenny Huang
khuang@intertrust.com