UNIVERSITY OF CAMBRIDGE

Introduction to Trusted Execution Environments (TEE) – IY5606

Steven J. Murdoch

Computer Laboratory

Learning objectives

Trusted Execution Environment (TEE)

- Understand what a TEE is and why it is of interest
- Appreciate the range of standards and products that offer TEE capability
- Be able to describe the basic building blocks of a typical TEE
- Compare the attack resistance of a TEE product w.r.t. security evaluated smart cards
- Contrast ownership and management issues w.r.t. a traditional smart card/SIM model

Trusted vs. Trustworthy

- Trusted
  - Someone or something **you rely upon** to not compromise your security
- Trustworthy
  - Someone or something **will not** compromise your security

- Trusted is about how you use something
- Trustworthy is about whether it is safe to use something
- **Trusted Execution Environments** are what you may choose to rely upon to execute sensitive tasks
- Hopefully they are **Trustworthy** too!

Goals of a Trusted Execution Environment

TEE [Vasudevan et al.]

- Isolated Execution
  - TEE may be malicious
- Secure Storage
  - Integrity, Confidentiality, Freshness
- Remote Attestation
- Secure Provisioning
- Trusted Path

Traditional security

- Confidentiality
- Integrity
- Availability

Example applications

- Cryptography
  - Key storage (never leaves the TEE in the clear)
  - Key usage policy enforcement
- Password verification
  - Commonly used to unlock keys
- Digital Rights Management (DRM)
  - Typically involves cryptography
  - Also requires control over peripherals

Trusted Computing Base

- The Trusted Computing Base is the smallest amount of code (and hardware, people, processes, etc.) that you must trust in order to meet your security requirements
- Confidence in the TCB can be increased though
  - Static verification
  - Code inspection
  - Testing
  - Formal methods
- All of these methods are expensive, so reducing the complexity of the TCB is an important goal but **is not sufficient**

Trusted Computing Base

- The Trusted Co
  hardware, peop
  your security re
- Confidence in t
  - Static verificat
  - Code inspecti
  - Testing
  - Formal metho
- All of these met
  TCB is an impo

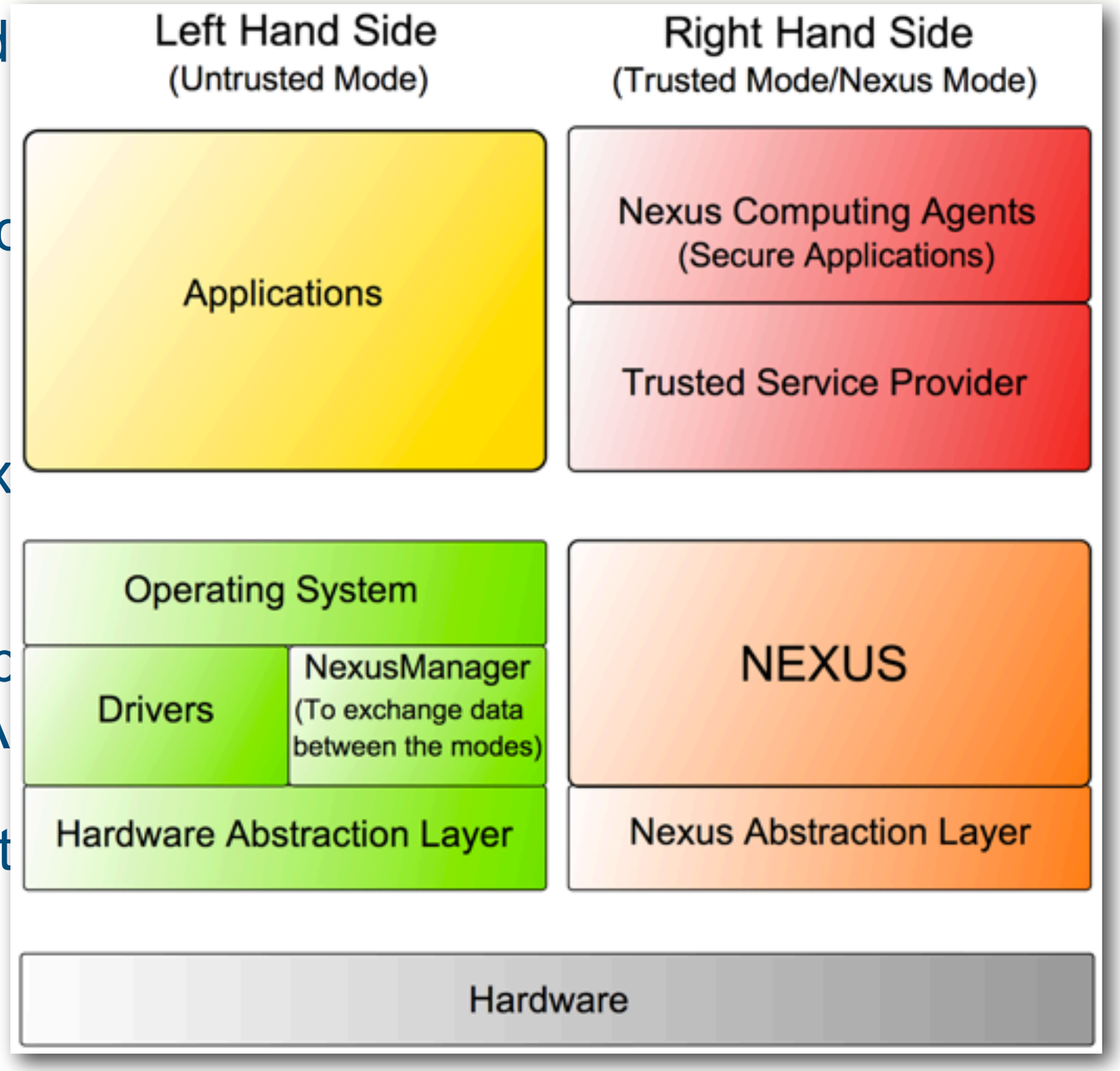
```
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa,
SSLBuffer signedParams, uint8_t *signature, UInt16 signatureLen)
{
    OSStatus        err;
    ...

    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;
    ...

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
}
```

Trusted Platform Module (TPM)

- It was hoped that the TPM could create a TEE through the **Static Root of Trust Measurement** (SRTM)
  - Measure hash all software loaded since BIOS
  - OS would perform isolation
  - Attempted by Microsoft as Next-Generation Secure Computing Base (a.k.a. Palladium)
  - Shelved in 2004, but some aspects remain e.g. in Bitlocker disk encryption and Early Launch Anti-Malware (ELAM)
- Problem was that TCB became too large and too dynamic
- Two “identical” computers could have different hashes

Trusted Platform Module (TPM)

- It was hoped that the TPM could **Trust Measurement** (SRTM)
  - Measure hash all software loac
  - OS would perform isolation
  - Attempted by Microsoft as Nex (a.k.a. Palladium)
  - Shelved in 2004, but some asp encryption and Early Launch A
- Problem was that TCB became t
- Two “identical” computers could



Dynamic Root of Trust Measurement (DRTM)

- Rather than trust everything since BIOS, reset CPU and start measuring from that point on
- TPM v1.2 added dynamic registers (17–23)
  - Set to -1 on boot
  - Can be reset by OS to 0
- Register 17 is special
  - Only set by calling SKINIT (AMD-V) or SENTER (Intel TXT)
  - Disables DMA, interrupts, debugging
  - Measures and executes Secure Loader Block

Hypervisor TEE

- DRTM was intended to allow loading of a hypervisor
  - e.g. Xen or VMWare ESX
  - Hypervisor loads and isolates virtual machines
  - TPM can attest to hash of hypervisor
    - TPM sealed storage can be released only to hypervisor once it has been loaded properly
    - Some optimism that this would help for cloud computing
- Hypervisor is still a huge amount of code to validate (Xen contains a full copy of Linux; VMWare is of similar size)

Flicker TEE [McCune et al.]

- Flicker takes advantage of DRTM but for much smaller amounts of code (Pieces of Application Logic – PAL)

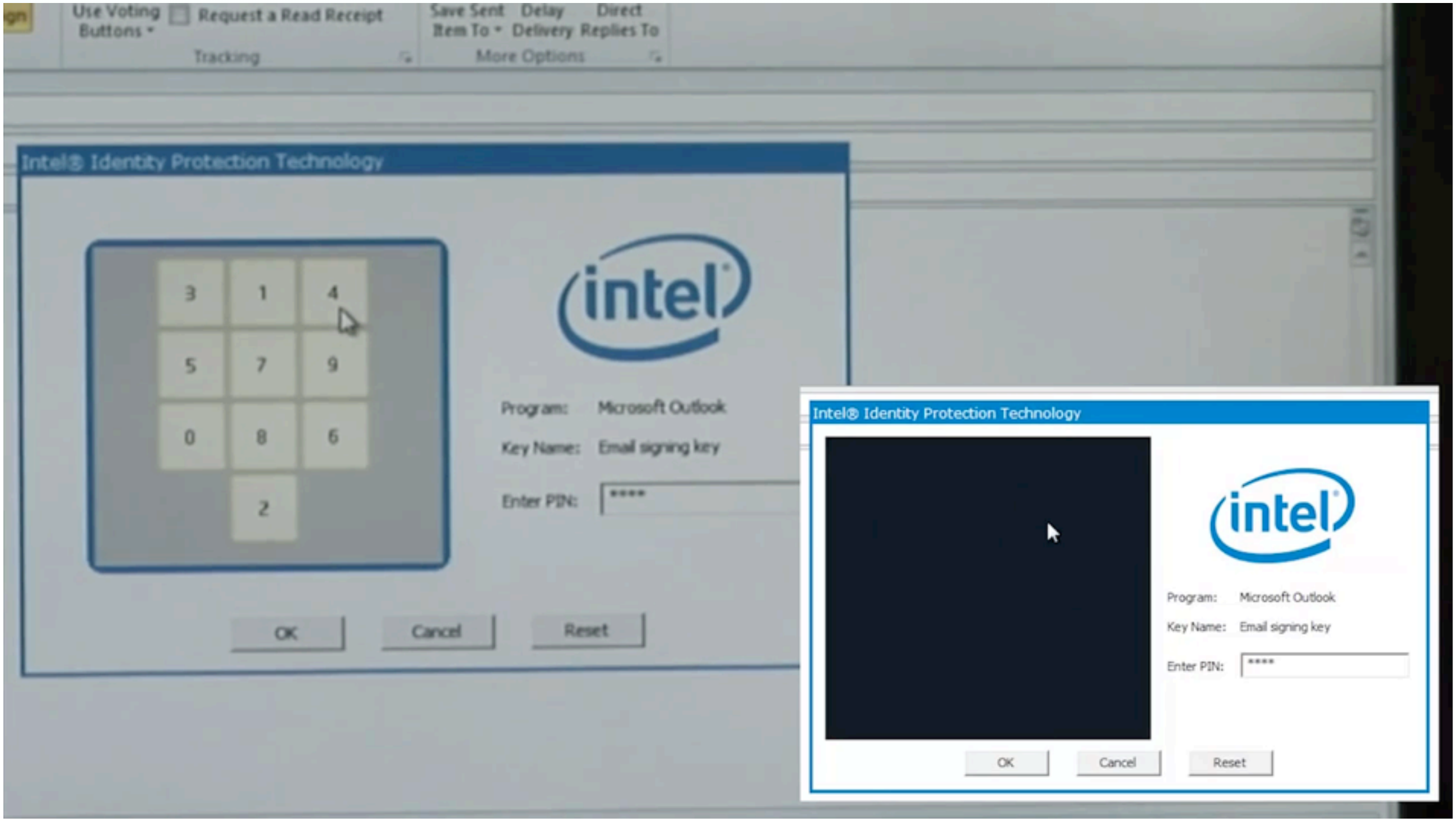
    1. Suspend OS
    2. Execute small amount of code on main CPU
    3. If necessary unseal storage and make changes
    4. Increment counter on TPM, storing this and data in sealed storage
    5. Restore OS

- Flicker runs at highest level of privilege so PAL is protected from OS but not other way around

Intel Identity Protection Technology [Carbin]

- Runs Java applet on separate CPU
  - Management Engine part of chipset so bound to physical hardware
  - ARC4 CPU
  - Applications currently available include
    - Key generation and storage (integrated with Windows Cryptographic API)
    - One time password generation (VASCO MYDIGIPASS.COM)
    - Secure PIN entry
      - Possible because chipset also manages video

Intel IPT video



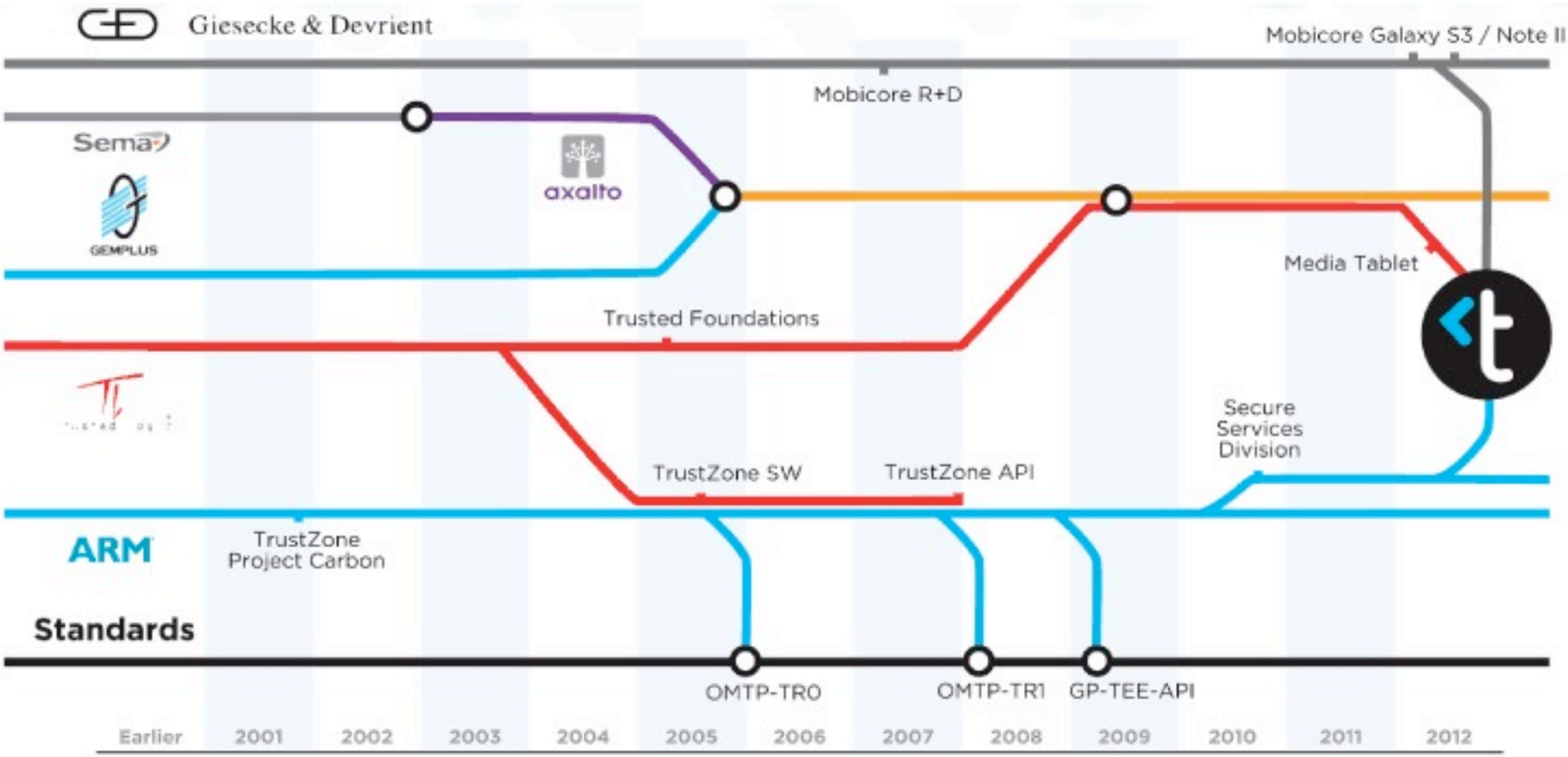
ARM TrustZone

- CPU buses extended to a “33rd bit”, signaling whether in secure mode
- Signal exposed outside of the CPU to allow secure peripherals and secure RAM
  - Potentially could have indicator for which mode CPU is in
- Open system and documented, but only allows one secure enclave

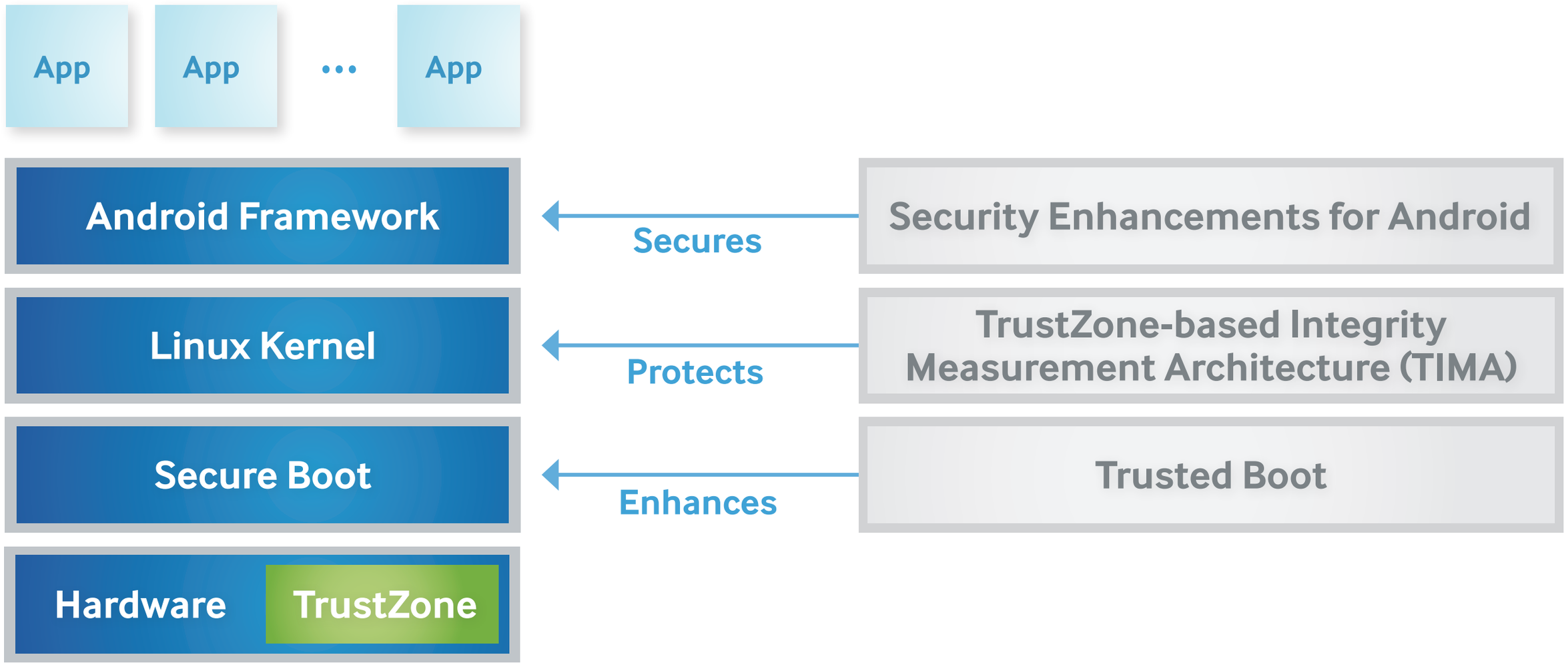
Trustonic

- TrustZone is not very useful by itself due to only allowing one enclave
- Gemalto developed the Trusted Foundations system
- G+D developed MobiCore
- Both split the one secure enclave into several, essentially through a smart card operating system
- Trustonic now developing based on MobiCore
- License fees required to implement code
- Samsung Knox is similar, but also introduces secure boot

Trustonic family tree



Samsung Knox [Samsung]



Intel SGX

- Tightly integrated with CPU
  - Modifies memory management
  - Enclaves protected from other code, and vice versa
- As enclaves are built, the code is measured in a similar way to the TPM
- Can be combined with IPT for trusted display
- Demonstration uses HDMI DRM for trusted path
- No publicly available hardware yet

Comparisons

- Economic
  - Lock-in
  - Who pays the license fees?
- Performance
  - What can you do in the TEE?
- Functionality and flexibility
  - How well connected is the TEE to the CPU and what can it do?
- Security
  - What attacks are feasible

Detour: Security Economics

- About 8 years ago, researchers started looking at the Economics behind security decisions
- Failures in security were not primarily a result of not enough cryptography, but due to failure of incentives
- Why should you protect your computer against malware when it will be a victim of a DDoS who pays the cost
  - c.f. why should you clean your sewage when it is the people downstream who bear the cost of pollution
- Supporting an economic model is now the primary goal of many security mechanisms
  - e.g. printer cartridge authentication

IT Economics is different

- Network effects
  - Value of a network grows super-linearly to its size (Metcalfe's Law says n^2, Briscoe/Odlyzko/Tilly suggest $n \log n$); this drives monopolies, and is why we have just one Internet
- High fixed and low marginal costs
  - Competition drives price down to marginal costs of production; but in IT industries this is usually (near as makes no difference) zero; hence copyright, patents etc. needed to recover capital investment
- Switching costs determine value
  - Switching from an IT product or service is usually expensive; Shapiro-Varian theorem: net present value of a software company is the total switching costs to the nearest competitor

Economic impact on TEE design

- Time to market is critical
  - High fixed/low marginal costs, network effects, and switching costs give first mover a big advantage
  - Security often doesn't help here and often hinders
- Appealing to complementers also important
  - People buy your product because of other products you enable
- Locking someone into your platform is valuable
  - Your customers know this too
- Regulation is often needed but normally too slow

Economic considerations

- Wide variety of business models in play, and these have as much to do with the design choices as security
- Open specification (TPM) vs closed specification (SGX)
- Platform specific (IPT) vs generic platform (TPM)
- License fees paid to
  - Chipset vendor (IPT)
  - Handset manufacturer (Samsung Knox)
  - OS vendor (Google NFC)
  - CPU core developer (Trustonic)

Performance

- TPM
  - TPM is really slow, and only has slow communications bus to CPU
  - Flicker TEE code runs on main CPU so can be as fast and has access to as much RAM as OS will spare
- Trustzone (Trustonic, Samsung Knox)
  - TEE code runs on main CPU so is as fast, but RAM may be limited
- Intel IPT
  - TEE code runs on separate CPU, which is moderately fast
- Intel SGX
  - TEE code runs on main CPU so is as fast, as much RAM as needed

Functionality

- TPM is isolated from CPU, so can only operate on what it is provided with (hence why Flicker and similar systems are needed)
- Trustzone only allows one compartment (hence why Trustonic and Knox are needed)
- IPT could be thought as similar to TPM (though used very differently)
  - Also has access to display (likely as a result of its ME heritage)
- SGX very tightly integrated into CPU
  - Has control over virtual memory management
  - Very fast context switching and high-speed communications

Flexibility

- TPM functionality baked into hardware
  - Designed to be flexible but what is there cannot be changed
- Flicker allows arbitrary code to be run, but it does not have access to OS or drivers
  - As a result only computation and very simple I/O possible
- Intel IPT, Trustonic, Knox can run arbitrary code
  - But it has to be licensed by Intel, Trustonic, Samsung first
- SGX allows anyone to run arbitrary code
  - But how will attestation key business model be managed?

Security: side channel attacks

- If malicious code can share the same CPU as the TEE there is a risk of side-channel communication
  - If goal is to separate two pieces of malicious code then covert channel communication is also a risk (but normally now)
- Examples include hyperthreading vulnerability in Intel CPUs
- Trustzone (Trustonic, Knox), SGX shares same CPU between TEE and untrustworthy code
- Intel IPT shares CPU with Intel-managed but possibly compromisable code
- TPM runs only security-oriented code which is (hopefully) well written and tested

Security: physical attacks

- Physical attacks on TEE generally considered outside of threat model
  - Very hard to defend against, not a scalable attack
- Leak of attestation keys could be a problem
  - DRM application cares whether code runs on a CPU or emulator
  - Revocation can help with this
- TPM comes from smart card world so likely has some physical protection of keys (but this has been found flawed in some cases)
  - Interface to CPU not protected at all
- Trustzone keys are in unencrypted flash
- SGX/IPT keys are on CPU which should be non-trivial to extract

Security: platform binding

- TPM chip is not well bound to CPU (could be removed or replaced)
- Intel IPT is more tightly integrated with CPU so much more challenging to remove or monitor communications
- Trustzone and SGX is the CPU so should be infeasible to modify without some serious hardware investment
- Good platform binding allows new types of applications (c.f. smart cards)
- Important to distinguish between scalable attacks
  - Break once, run anywhere
  - Broken until revoked
  - One device at a time

Comparison with smart cards

- Performance
  - TPM similar
  - SGX, TrustZone, IPT much faster
- Flexibility
  - TPM less flexible
  - Trustzone, SGX better (IPT, Trustonic, KNOX similar)
- Security
  - TPM similar (IPT too?)
  - Trustzone, SGX, Trustonic, KNOX likely less secure

TEE goals

	Isolated Execution	Secure Storage	Remote Attestation	Secure Provisioning	Trusted Path
TPM	Not really (too limited)	Yes (but very limited)	Yes	Yes	No (easily bypassed)
Flicker	Yes (but no drivers)	Yes (though TPM)	Yes (through TPM)	Yes (through TPM)	Limited
Trustonic/ KNOX	Yes (but restricted)	Yes	Yes	Yes	Somewhat
IPT	Yes (but restricted)	Yes	Yes	Yes	Somewhat
SGX	Yes	Yes	Yes	Yes	Probably

Conclusions

- TEEs may offer Isolated Execution, Secure Storage, Remote Attestation, Secure Provisioning, Trusted Path
- Wide variety of products available which fulfill these goals to varying extent
- Design choices affect both what TEE applications can be supported and how secure they are
- Smart cards may offer better security, but lack of platform binding makes some applications infeasible to support
- Economics will have as much to do with design choices adopted then security
- Ownership of the platform is a key difference between different solutions

References

- A. Vasudevan, E. Owusu, Z. Zhou, J. Newsome, and J.M. McCune. Trustworthy Execution on Mobile Devices: What security properties can my mobile platform give me? In Trust and Trustworthy Computing, vol. 7344 of LNCS, pp 159–178. Springer, 2012.
- Jonathan M. McCune, Bryan Parno, Adrian Perrig, Michael K. Reiter, and Hiroshi Isozaki. Flicker: An Execution Infrastructure for TCB Minimization. In Proceedings of the ACM European Conference on Computer Systems (EuroSys'08), Glasgow, Scotland, 31 March–4 April 2008.
- Paul Carbin. Intel® Identity Protection Technology with PKI (Intel® IPT with PKI). Technology Overview. 22 May 2012
- Samsung. White Paper: An Overview of Samsung KNOX. Enterprise Mobility Solutions. September 2013.
- B. Briscoe, A. Odlyzko, B. Tilly. Metcalfe's Law is Wrong, IEEE Spectrum, July 2006, pp. 26–31.
- Carl Shapiro, Hal R Varian. Information Rules: A Strategic Guide to the Network Economy. December 1998