# MODERN CRYPTO SYSTEMS AND PRACTICAL ATTACKS

DR. NAJWA AARAJ – SVP, SPECIAL PROJECTS

INTERNATIONAL CRYPTO MODULE CONFERENCE
18 MAY 2016

DARKMATTER

GUARDED BY GENIUS

# COMPUTATIONAL SYSTEMS EVOLVED …

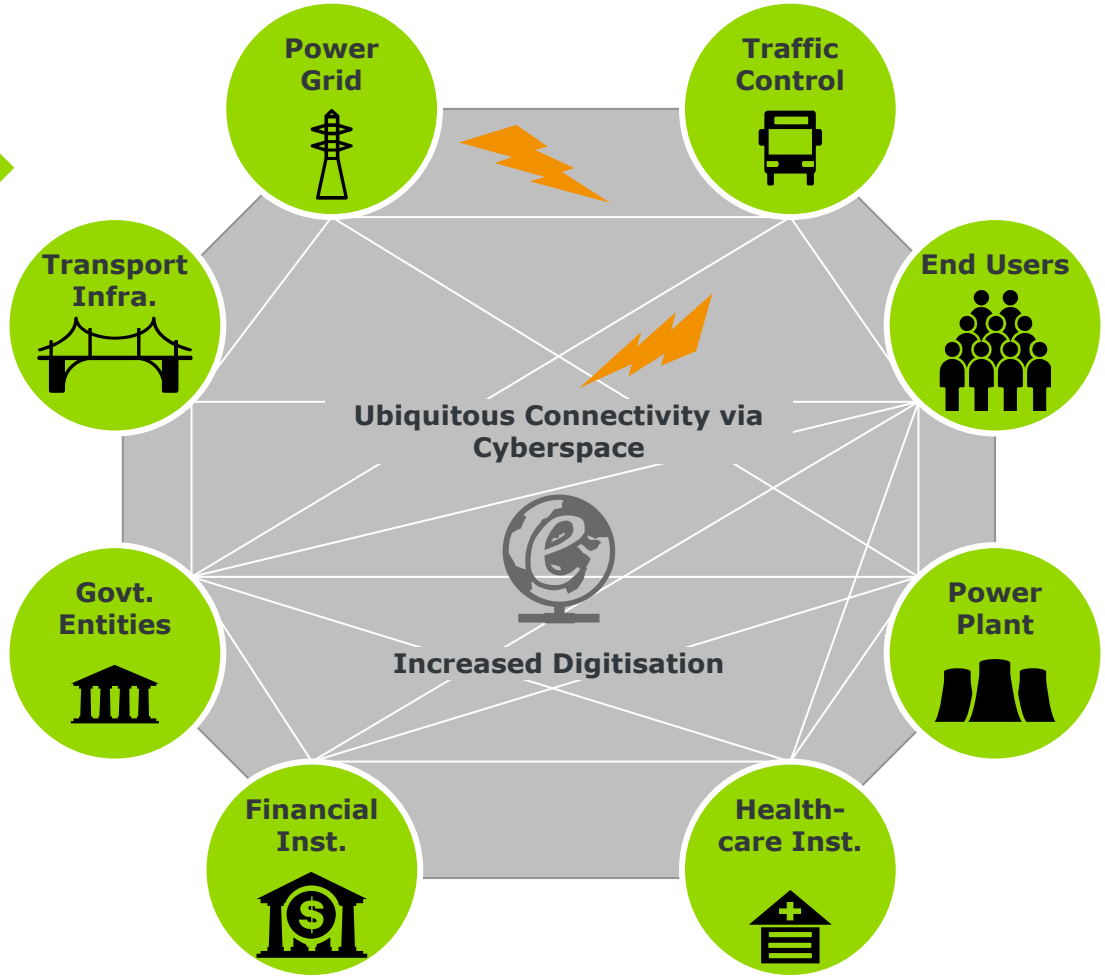
1970 | 1980 | 1990 | 2000 | 2010 | 2016+

**Mainframes**
- Computers to process large amounts of data

**Desktop and internet**

**EMBEDDED SYSTEMS AND Ubiquitous computing**

**Cyber Physical Systems (10' s)**



# … AS DID ATTACK SOPHISTICATION

**Cyber Attacks Types**

**Attack Type** → Classifies attacks mainly based on level of expertise and effect control

| | Target | Required Expertise Level | Effect Control | Purpose | Attack Vectors |
|---|---|---|---|---|---|
| **Simple Unstructured Attacks** | • Single connected user<br>• Consumer with information of low sensitivity/public information | • Low | • Unfocused | • Disruption<br>• Small scale theft and fraud | • Cyber fraud<br>• Malware Outbreak |
| **Advanced Structured Attacks** | • Business Heads<br>• Executives dealing with restricted information | • Medium | • Basic | • Disruption of Services<br>• Espionage<br>• Large scale theft and fraud<br>• Destruction of information | • Cyber fraud<br>• Cyber extortion<br>• Malware and Rootkit Outbreak at OS and User level |
| **Complex and Coordinated Attacks** | • Heads of State<br>• Government Leaders<br>• Executives dealing with highly sensitive [top secret and secret] data | • Very high | • Sophisticated | • Disruption of Services<br>• Espionage<br>• Large scale theft and fraud<br>• Destruction of information<br>• Terrorism | • Sophisticated Malware and Rootkit Outbreak at OS and User level<br>• Advanced code manipulation<br>• Cyber theft<br>• Cyber espionage |

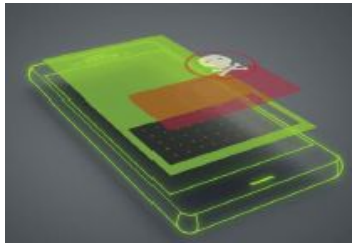
# WHAT WE DO

## What we do

- Robust and Efficient Cryptographic Protocols
- Research in Cryptography and Cryptographic Implementations
- Key Management and Trusted Platforms Research
- Operating Systems Security Research
- Frameworks for Secure Communication and Practical Encryption

## Our aim

- Secure and Authenticated Communication Channels
- Data at Rest and Data in Transit Security
- Operating Systems and Software Stack Security
- Hardware Security





# SECURE COMMUNICATION SOLUTIONS ARE REQUIRED

## Communication Security

- E2E Secure Communication transmitted over Voice, SMS, data, and Video Network
- Top Secret Cryptographic Algorithms
- Strong Crypto implementations with Resilience to Side Channel Attacks and other types of Cryptanalysis
- Hardware Rooted Key Management
- Improved EMM and RNG

## OS and Kernel Security

- Security Extensions – OS and kernel levels
- Integrity Monitor
- Process Isolation and Type Enforcement
- Secure Boot
- MMU Security
- Full Encryption of Data at Rest
- Hardware-based Root of Trust


## Management

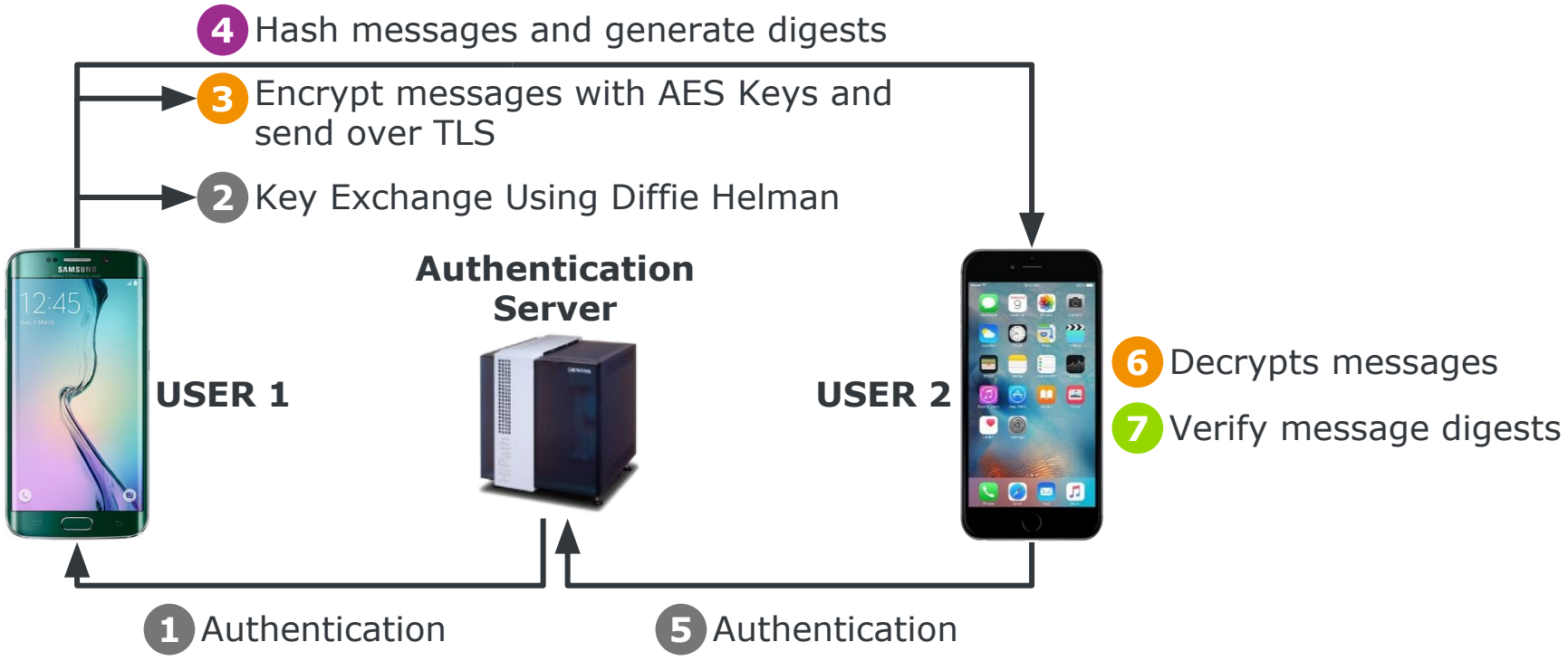
- Over the Air Device Management
- Auditing of Critical Events
- Centrally Managed Policies

## Infrastructure Security

- On-premise deployment within local infrastructure residing on secure military grade security infrastructure (secured at the highest national level)

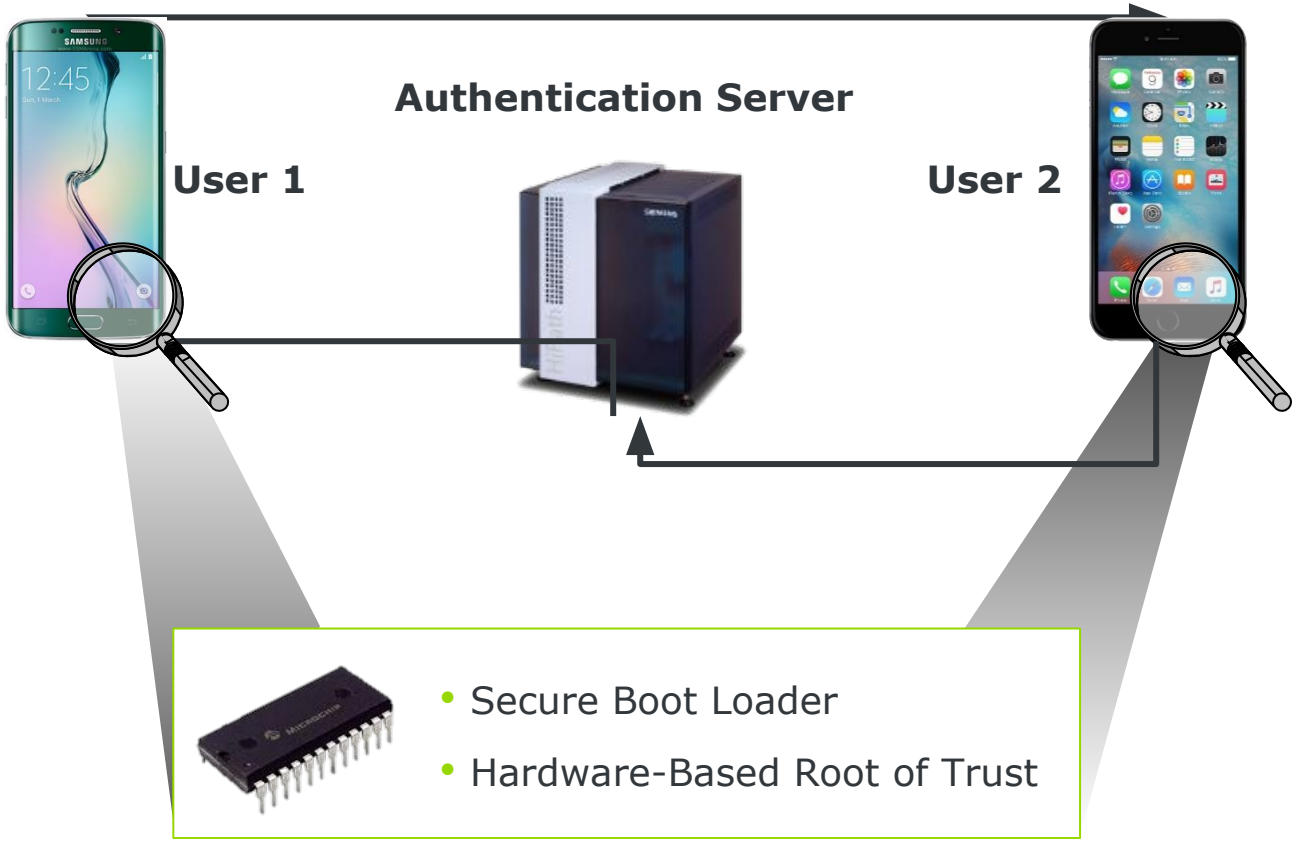
# SECURE PROTOCOLS ARE NEEDED …

X Public Key Crypto  X Private Key Crypto  X Hashing




- Top Secret Cryptographic Algorithms
- Hardened Cryptographic Library/ EMM/ RNG
- Robust Authentication and Localized PKI
- Perfect Forward and Future Secrecy
- Anonymity vs. Non-Repudiation

# … SO IS KERNEL AND HARDWARE LEVEL SECURITY




- Data at Rest Encryption
- Real-time Integrity Monitor
- Hardened Cryptographic Library
- Hardware-Rooted Key Management
- Hardened OS and Kernel

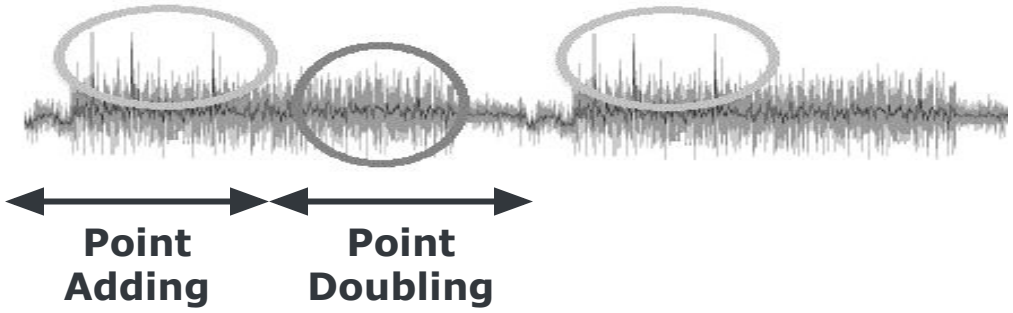
# CRYPTO POSES CHALLENGES FOR COMMUNICATION SOLUTIONS

- Encryption should be intractable by **theoretical (Crypt)-analysis**
  - E.g., Intractable properties of factoring large numbers

- Systems can be broken by **inadequate bit-level security and crypto parameters**
- **Inadequate Root of Trust** for the Cryptographic System
- Systems can be weakened by **inappropriate implementations**
  - E.g., **side-channel attacks;** memory attacks

**"By Attacking a Crypto Device, the adversary hopes to subvert its security correctness properties by extracting some secret"**

# PRACTICAL EXAMPLES




- Power consumption depends on computation performed
- Point multiplication (K*Q) occur during key and signature generation (Double-and-add algorithm)
- double and add are very different ops (key bit ==1)
- Simple Power Analysis (SPA): Leakage during point multiplication leaks information about secret (K)

- Power Analysis on Smart Cards – Retrieval of smart card PINs
- Smart Phones – Leakage of keys used to secure communications, financial transactions, SSL traffic, data stored on phones

## Most Common Side Channel Attacks

- **Power analysis –** A type of side channel attacks has been demonstrated in many cryptosystems; emanate from circuits which typically consume differing amounts of power based on operations and data
- **EM** – Electromagnetic emissions and signals (e.g., through near-field inductive and capacitive coupling, or far field antennas)
- **Timing** – Data-dependent differences in process time and delay in cryptographic operations based on data input and relevant operation

# TYPICAL TARGETS

- **Most Common Targets:**
  - Smart Cards / FPGAs - Microcontrollers/ Custom Processors
- **Other Common Targets:**
  - Smart Phones / Embedded Devices / CPU Boards
  - USBs / Consoles /
- **Modern Targets**
  - Charging patterns of devices
  - Interfaces of software / hardware co-design interactions

- **When interception is feasible (active / passive), simple or differential side channel attacks will aim at breaking modern crypto**

# COUNTERMEASURES

- **Algorithm-level Countermeasures**
  - Randomness (masking / blinding)
  - Constant Time implementations
  - Pre-computations and Leak Reduction
  - Noise based countermeasures
  - **Increase dependencies on Boolean ops (e.g. keecak)**
  - **Randomize in-algorithm structures between rounds**
- **Protocol level countermeasures**
- **Architecture-level Countermeasures**

# PROTOCOL LEVEL COUNTERMEASURES

- **Intent:**
  - Reduce the amount of leakage to less than the minmum required for key(s) recovery using SPA / DPA / EM-based leakage
  - Reduce interim states that could lead to leakage
- **Countermeasures:**
  - Key Agility (per session / per call / per message)
  - Layered Security
  - Increased Overall bit level security
  - Redundant crypto operations to reduce leakage and temp values
  - Smart choice of the cryptosystem

- Leakage introduced by Client – Server protocols
  - Protocol execution split into algorithm (r_j) and protocol rounds (R_j) and leakage function defined during and after execution of r_j and R_j
  - Unbounded (continuous) computational-leakage: impact of leakage throughout overall cryptographic protocol execution
  - Challenge of finding Lmin
- Leakage introduced through authentication Process
- Master / Ephemeral key tree / structure
  - Internal State / Leakage and impact on data confidentiality and Privacy on client and server sides
- Leakage / Faults introduced through multi-threaded crypto

# ARCHITECTURE-LEVEL COUNTERMEASURES

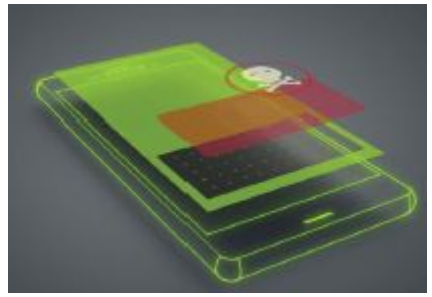
- **Intent:**
  - Reduce the amount of leakage to less than the minimum required for key(s) recovery using SPA / DPA / EM-based leakage through architectural / logical modifications on the underlying system
  - Manipulate interim structures (physical and logical) that could lead to leakage
- **Countermeasures:**
  - Hardware / software co-design
  - Hardware-based crypto acceleration
  - Gate-level modifications
  - Chip-level partitioning
  - Tamper Proofing and physical security

- Leakage resources across all different components interfaces. Software / hardware co-design architecture:
  - Crypto cards / TPM integration of an embedded system SoC
  - Crypto Accelerators and Security modules on / off Chipset
- Leakage overheads imposed through architectural functions in end applications
  - Power signals, EM signals, memory leakages,
  - Linear and differential properties of such signals in isolation and through their correlation
- Leakage in uniprocessor systems as well as within latent parallelism of multiprocessor systems

# SECURE COMM. SOLUTION – RECOMMENDATIONS TO CONSIDER

**Adopt the Right Secure Communication Solution Across User/OS/Kernel space and Hardware**

- Protocol level countermeasures
  - New leakage areas are introduced?
- Properly defined Root of Trust for your system and Right balance between Software and Hardware
  - New leakage areas are introduced?
- Solutions with Built-in Countermeasures within your code …
  - Is implementation cryptographically secure (DPA, SPA, memory attacks, etc.)?
- Strong OS and Kernel-level hardening
- Right bit-level of security and right crypto parameters




# SECURE COMMUNICATION – OUTLOOK

**Technology Advancement**

- Attacks Sophistication will Only increase
  - Using secure communication and email applications is important
  - However – Security should be boasted through **integrated and secure Hardware, Firmware, Hardened Operating system, Software apps** and **services**
  - Hardware-Owned Roots of Trust
- On-premise/in-country solutions management, development and provisioning

**Protocol Advancement**

- Resource-Restricted Protocol Undifferentiability
- Requirement for Perfect (Conditionally-Perfect) Secrecy and Anonymity-preserving Protocols

**Computational Complexity**

- Quantum information processing systems?

# THANK YOU

DARKMATTER