# Security Models & Covert Channels

ECE568 – Lecture 18

Courtney Gibson, P.Eng.
University of Toronto ECE

# Outline

**Security Policies**

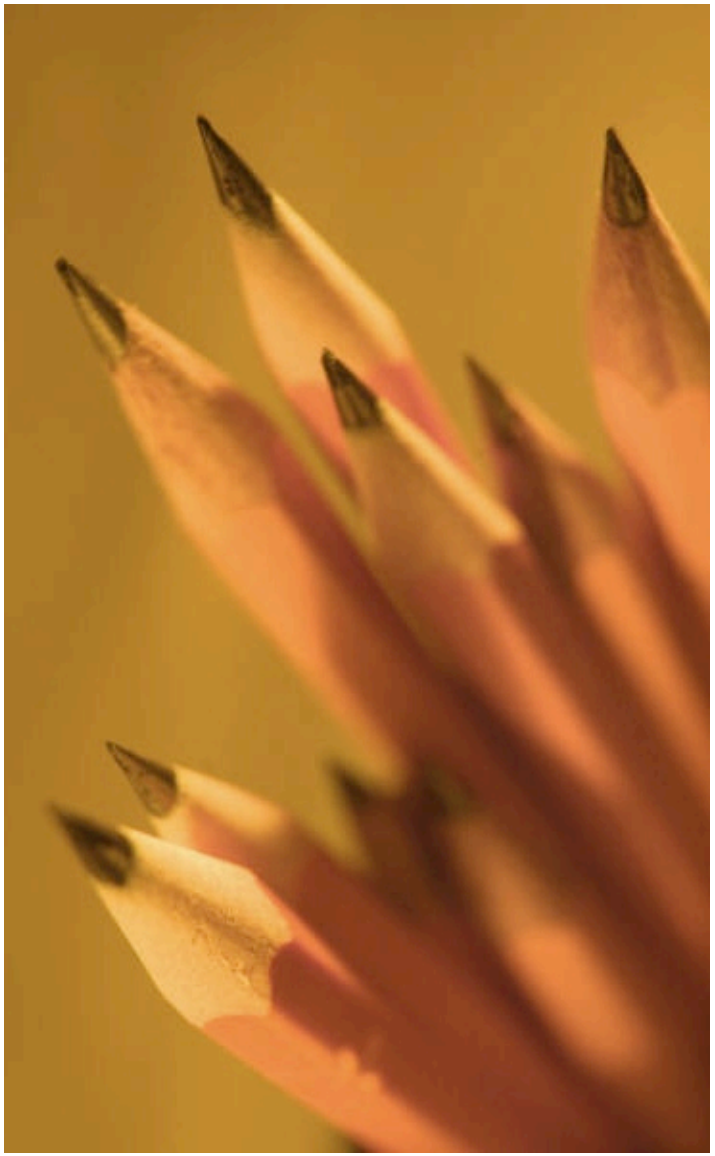
- The Bell-LaPadula Security Model
- Biba Integrity Model
- Chinese Wall Model
- Clark-Wilson Model

**Covert Channels**

- Types of covert channels
- Non-interference

**Certification Schemes**

- Example: Orange Book
- Example: Common Criteria



# Security Policies

# Security Policies

**Security policies** govern how a system handles information

- Their goal is to ensure that a system maintains some security property
- They are mainly used by the military, government, etc.

Security policies are based on **security models** that are generally quite intuitive, and yet rigorous enough that they stand up to proofs for security

- These models help check if a system is designed securely
- The system will be secure if there are no bugs

Security policies are implemented as mandatory access control (MAC) policies

- They work with or trump any user-specified DAC policies

# Types of Policies

**Confidentiality Policies**

- Confidentiality defines who is authorized to access information or resources
- These policies help protect information from leaking
- Historically, they are used for military applications
  - Protect secret information from being leaked to the enemy
  - Companies can protect customer privacy or corporate secrets

**Integrity policies**

- Integrity defines the trustworthiness or reliability of information
- These policies help prevent corruption of information
- Normally, they are used by companies
  - Protect destruction, defacement of company data

# Bell-La Padula (BLP) Model

The Bell-La Padula Model or Multi-level Security Model (MLS) is used to build confidentiality policies

- It has two types of **elements** in the system
  - **Subjects** are active participants in the system
  - **Objects** are data or resources that need to be protected
- There are various **classification levels** defined by the system, ordered from highest to lowest
  - *e.g.,* Top Secret (TS), Secret (S), Classified (C) and Unclassified (UC)
- There are a number of **categories** in the system
  - *e.g.*, FBI, NSA, President (POTUS)

# Bell-La Padula (BLP) Model

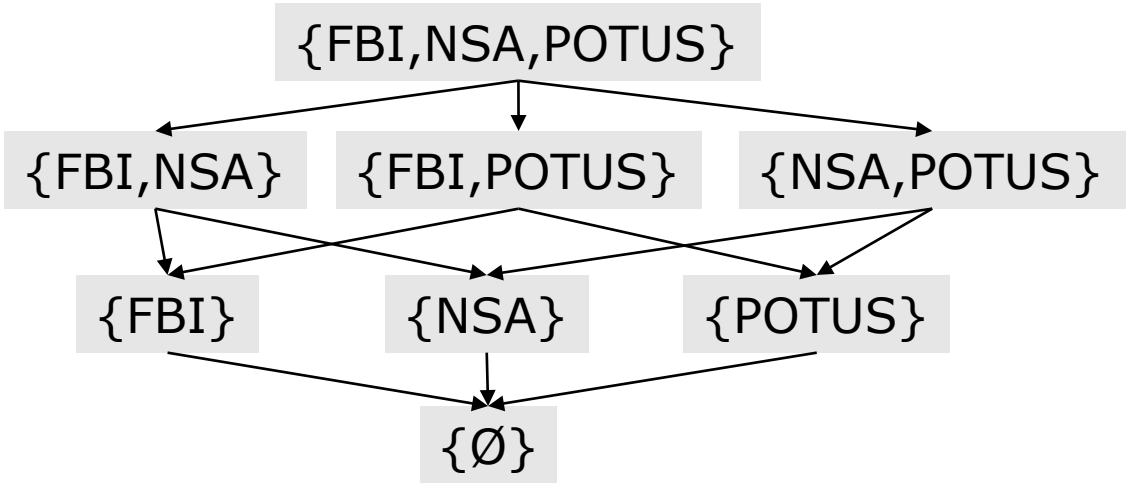
Each object is associated with a **security level** of the form (classification level, set of categories)

- *e.g.*, {TS, {NSA, POTUS}} means this object contains top secret information pertaining to NSA and the President

- Each subject has a **security clearance** up to a given classification level, and their maximum security level is defined as (security clearance, set of categories)
  - *e.g.*, a subject may be assigned the {S, {NSA}} security level

# Category Set Lattice

The set of categories forms a **lattice** (a partially ordered set) with a **dominance** relation




If element $E_1$ has a higher security clearance than element $E_2$, and $E_1$'s category set contains $E_2$'s category set, then $E_1$ is said to **dominate** $E_2$

- *e.g.*, {TS, {NSA,POTUS}} dominates {S,{NSA,POTUS}, {TS, {NSA}}, etc.

# Bell-La Padula Security Model

Bell-La Padula (BLP) defines two rules:

- **Simple Security Property:** A subject **s** can read an object **o** iff the security level of **s** dominates security level of **o**, and **s** has discretionary read access to **o**
- ***-Property (Star Property): s** can write **o** iff the security level of **o** dominates security level of **s**, and **s** has discretionary write access to **o**

- These rules say you can only read from things less secure than you (*i.e.*, no read up) and write to things more secure than you (*i.e.*, no write down)
- Information can **never** flow from a more secure (confidential) source to a less secure one
- In practice, exceptions require "trusted subjects"

# Biba Integrity Policy

Bell-La Padula does not provide any integrity, because a low security level subject can modify high security level objects

- The Biba Model is used to build integrity policies
  - It prevents low integrity information from affecting higher integrity information
- Biba is the mathematical dual of Bell-LaPadula
  - Each Subject and Object has an associated integrity level
  - Also, subjects can execute other subjects (*i.e.*, programs)

# Biba Integrity Policy

Biba has three rules:

- **Simple Integrity Property: s** is permitted to read an object **o** iff the integrity level of **s** is lower than the integrity level of **o**, i(s) ≤ i(o) (*i.e.*, no read down)
- ***-Integrity Property:** A subject **s** is permitted to write to an object **o** iff the integrity level of **o** is lower than the integrity level of **s**, i(o) ≤ i(s) (*i.e.*, no write up)
- **s1** can execute **s2** iff i(s2) ≤ i(s1)

# Hybrid Policies

Hybrid Policies protect both confidentiality and integrity

- The **Chinese Wall** model is taken from British laws governing conflict of interest
- In the CW model, there are a set of objects
  - Each object belongs to a dataset
  - Each dataset belongs to a **Conflict of Interest (COI)** class
- A subject is allowed to read an object if either
  - The subject has previously read an object from the same dataset as the current object
  - The subject has not previously read any object from the COI class of the current object

# CW Policy Example

Consider a lawyer who works with banks and software companies:

- In this case, a company represents a dataset
- COI classes are defined by sets of competing companies
- Example:
  - Bank COI class = {TD, BMO, CIBC}
  - Software COI class = {IBM, Oracle, Microsoft}
- If a lawyer has worked for TD, she cannot work for BMO or CIBC, because that would be a COI
- However, she is free to work for IBM, Oracle or Microsoft
- If she starts working for IBM, then she can no longer work for Oracle and Microsoft
- What about another lawyer working for TD and Oracle?

# Clark-Wilson Policy Model

In 1987 David Clark and David Wilson introduced a model for analyzing the security of a commercial computer system

- Clark-Wilson uses two main concepts that provide the basis for an integrity policy
  - Data should only be manipulated in constrained ways using **well-formed transactions** that transform one valid state of the system into another valid state
  - The principle of **separation of duty** requires that the **certifier** of a transaction and the **implementer** be different entities

# Clark-Wilson Policy Model

Based on these concepts, Clark-Wilson defines two types of data and two types of actions:

- **Data**
  - **CDI (Constrained Data Items):** Items that have a format that they must follow
    - Example: an e-mail address must have an "@" sign
  - **UDI (Unconstrained Data Items):** Typically, user input
- **Actions** (Procedures)
  - **IVP (Integrity verification procedure):** Ensures that CDIs conform to the specified format
  - **TP (Transformation procedure):** Transaction that takes UDI/CDI as input and produces a CDI, and enforces the integrity policy of the system

# Clark-Wilson Policy Rules

Clark-Wilson enforces integrity by defining two types of rules that a secure system should follow:

- **Certification rules:** govern what certifiers check, i.e., whether the IVP and TP are written correctly (help administrator monitor integrity)
- **Enforcement rules:** govern what rules the system will enforce (help system preserve integrity)

# Clark-Wilson Policy Rules

**Rules**

- **CR1:** The system will have an IVP for validating the integrity of any CDI
- **CR2:** When a TP is run on some set of CDIs, it must transform valid CDIs into other valid CDIs

The rules above require that TPs are certified to operate on specific CDIs, hence the rules below

- **ER1:** System must maintain a list of certified relations, and ensure only TPs certified to run on a CDI change that CDI
- **ER2:** System must associate a user with each TP and a set of CDIs. The TP can access the CDIs on behalf of the user if the CDI is associated with the user.

# Clark-Wilson Policy Rules

ER1 and ER2 require keeping track of triples (user, TP, {CDIs}) called "allowed relations"

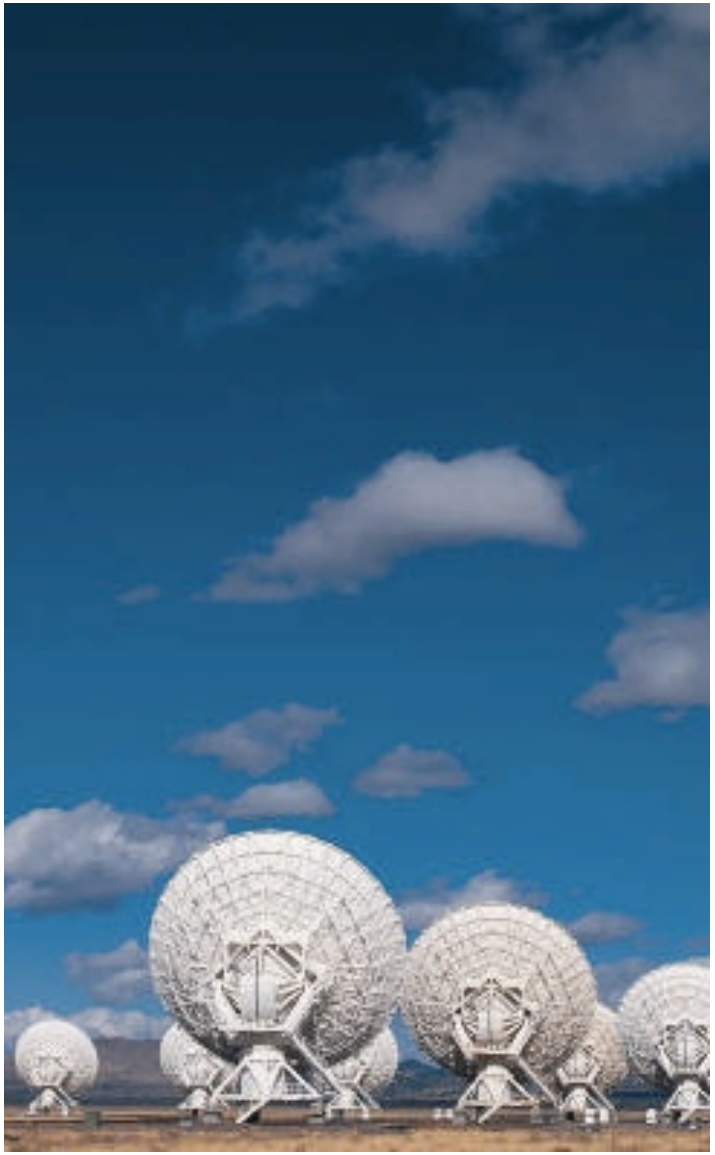
- **CR3:** Allowed relations must meet the requirements of "separation of duty"

We need authentication to keep track of CR3

- **ER3:** System must authenticate each user accessing a TP

More Rules

- **CR4:** All TPs must append their actions to a log (logging)
- **CR5:** Any TP that takes a UDI as input must either reject the UDI, or transform it into a CDI (validate all input)
- **ER4:** Only the certifier of a TP can change the list of users that can execute the TP. However, the certifier cannot execute the TP (separation of duty)



# Covert Channels

# Covert Channels

The previous security models specify how information should flow in the system when communicating **explicitly** (*e.g.*, read and write)

- A **covert channel** allows **implicit** information flow, outside of the specified policy
  - *e.g.*, suppose two processes share a lock on the file, then one process can transmit one bit of information to the other process, by holding or not holding the lock
  - *e.g.*, a process can vary the amount of time it takes to execute by causing many page faults, thus transmitting information to another party

Covert vs. Side Channels

- Covert channels usually imply that information is sent intentionally, and the sender wishes to remain undetected
- With **side channels**, information is leaked unintentionally

# Types of Covert Channels

Types of covert channels

- **Timing channel:** Information is transmitted by varying the time it takes to perform an action
- **Storage channel:** Information is transmitted via properties of a storage object (*i.e.*, size of a file, locks on a file, existence of a file, name of a file, etc.)
- **Termination channel:** Information is transmitted depending on whether a process has terminated
- **Power channel:** Information about what a circuit is computing is transmitted depending on the current drawn
- **Steganography:** Information is hidden within some content, such as an encoded text file within a picture

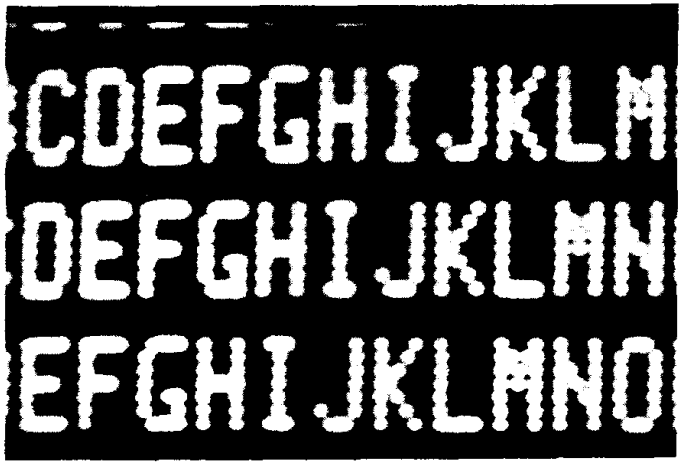
# Types of Covert Channels

Types of covert channels

- **Fault Injection Attacks:** Supply voltage, clock, temperature, radiation, etc.
- **Electromagnetic channel:** Analysis of radio emissions, etc.
- **Video channel:** Analysis of reflected light emissions

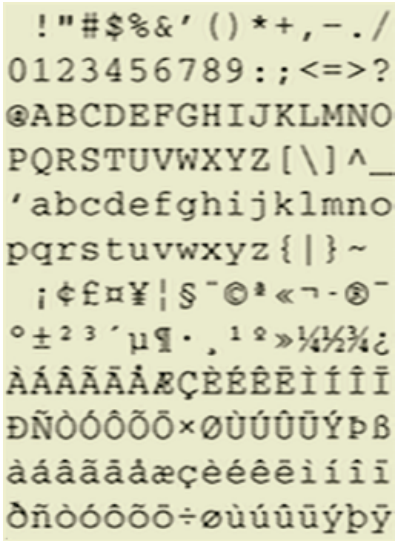
# Electromagnetic Side-Channel

- Wim van Eck published a seminal paper in 1985, examining the possibility of using stray signals from monitors, in order to reconstruct confidential video signals



# Electromagnetic Side-Channel

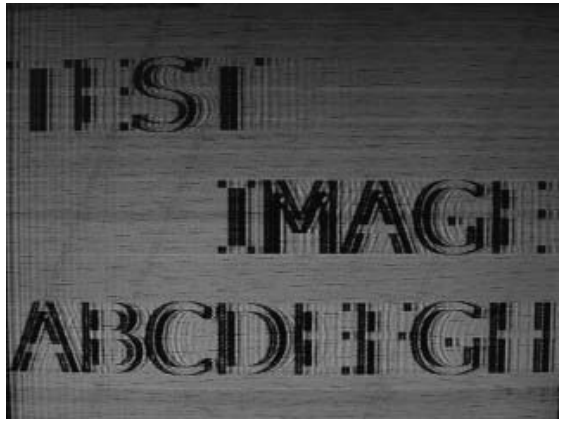
- Led to military TEMPEST program: eliminate potentially-compromising EM emissions
- Development of "TEMPEST fonts"




**Reference:** *Evaluation and Improvement of the Tempest Fonts*, Tanaka et al., Information Security Applications 2005

# Electromagnetic Side-Channel

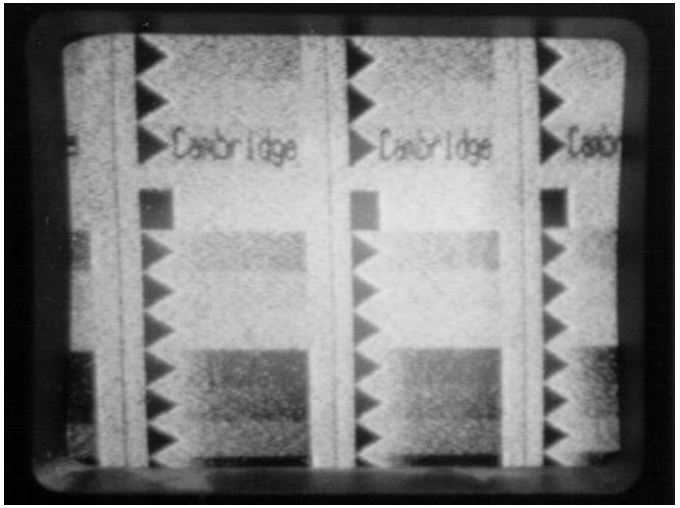
- Original work was done on CRT (Cathode Ray Tube) screens, but current works shows that this is still possible, using timing channel of modern LCD monitors

**Reference:** *Synchronization Clock Frequency Modulation Technique for Compromising Emanations Security*, Watanabe et al., EMC'09/Kyoto
http://www.ieice.org/proceedings/EMC09/pdf/21P1-4.pdf

# Electromagnetic Side-Channel

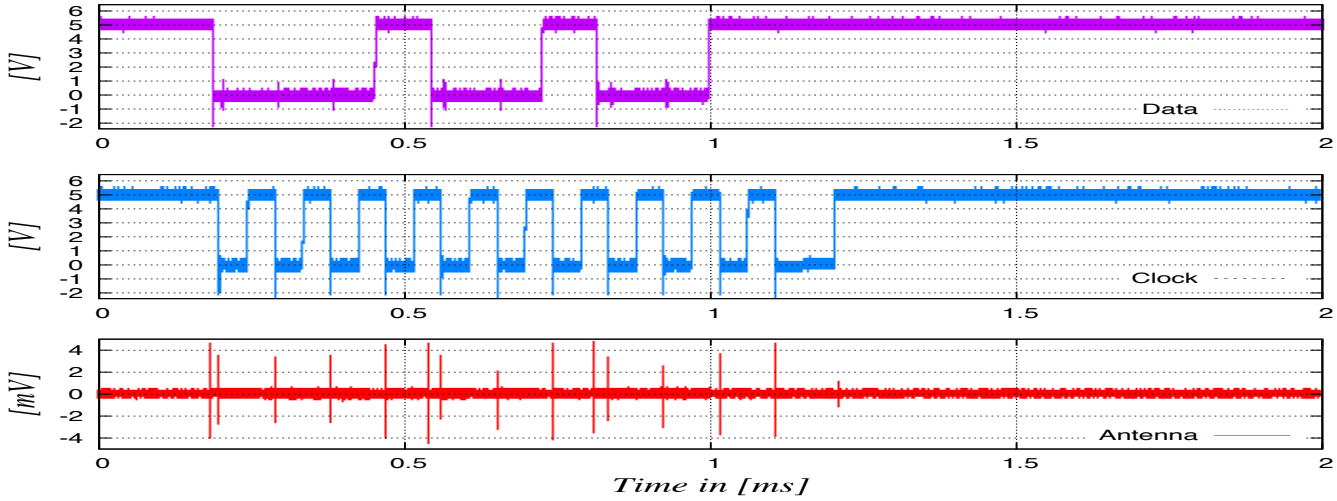
- Interesting military research in late 1990's on the feasibility of using TEMPEST to spread disinformation to an attacker


**Reference:** *Soft Tempest: Hidden Data Transmission Using Electromagnetic Emanations*, Kuhn et al., Workshop on Information Hiding (1998)
http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA389250

# Electromagnetic Side-Channel

- Same principle can be applied to other I/O devices (*e.g.*, keyboards)




**Reference:** *Compromising Electromagnetic Emanations of Wired and Wireless Keyboards*, Vuagnoux et al., USENIX 2009
http://www.usenix.org/events/sec09/tech/full_papers/sec09_attacks.pdf

# Non-Interference

Covert channels are generally very hard to detect

- They exist whenever the actions of one process **affect** the actions of another process in some way, even though there is no explicit communication

**Non-interference** allows analyzing convert channels

- A system has the non-interference property if and only if any sequence of inputs to a process will produce the same outputs, regardless of inputs to another process

For example:

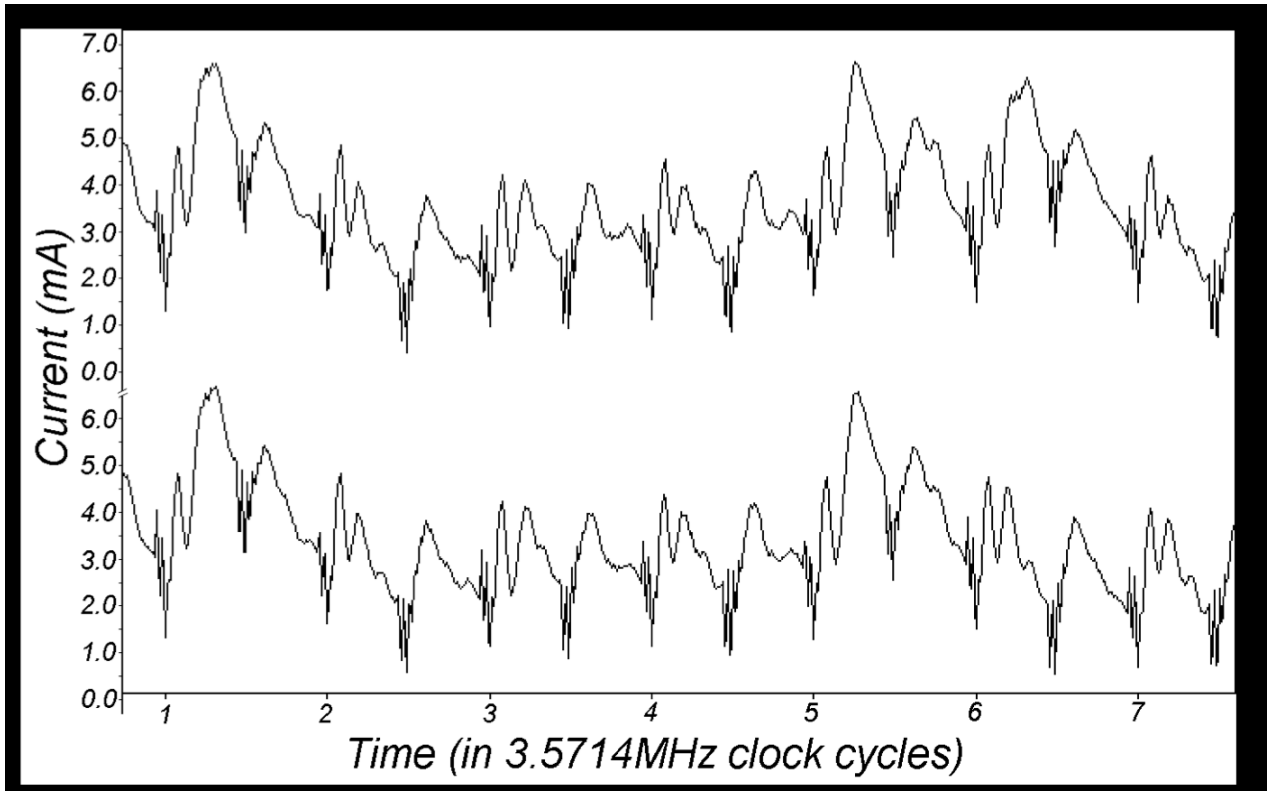
- Say a user with a 'low' clearance is working on a machine
- The machine will respond in the same way whether or not a user with a 'high' clearance is working on the machine
- Thus, the low user will not be able to acquire any information about the activities of the high user

# Non-Interference

Non-interference is a very strict property and it is difficult to make a system satisfy this property

- However, every covert channel identified using non-inference analysis is not necessarily usable
  - The sender side must be able to control what is transmitted into the channel, and the receiver must be able to interpret the information that is sent
  - The channel must have sufficient **capacity** to be useful, *e.g.*, a channel may not be useful if only one bit of information can be transmitted every hour

# Differential Power Analysis




**Reference:** *Differential Power Analysis*, Kocher et al., CRYPTO'99
ftp://amusing.mit.edu/afs/net/dev/admin/www/root/org/a/auto-id/foley/Reference/kocher99diffpowatk.ps

# Timing Analysis

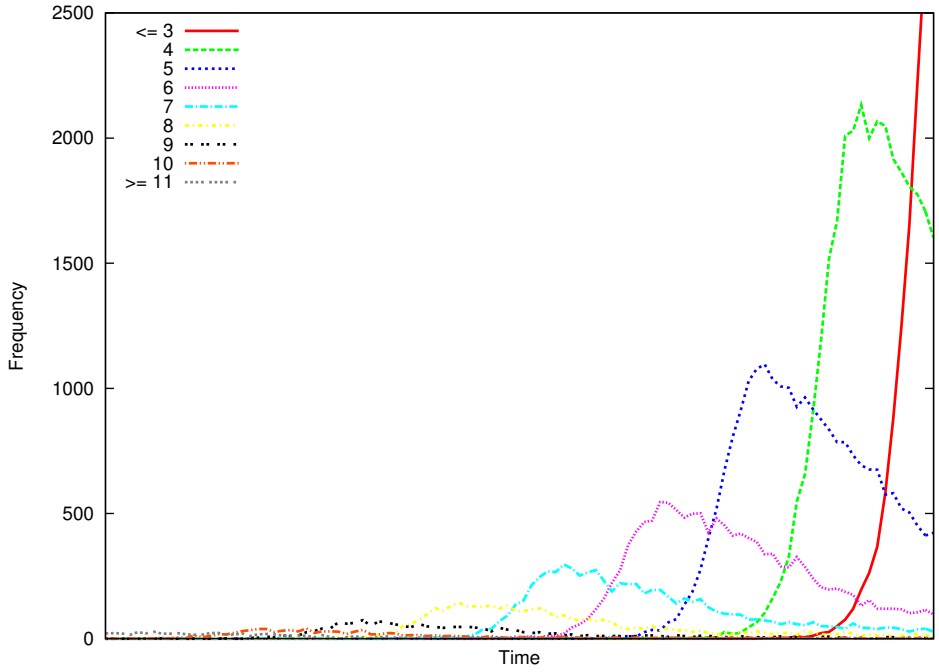
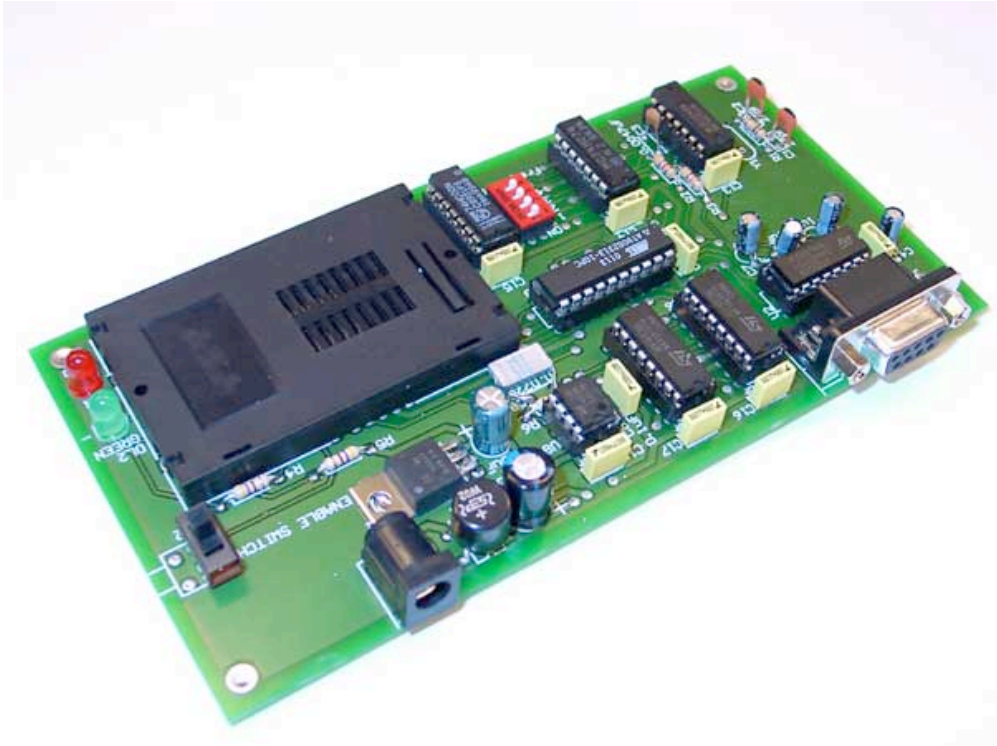



**Fig. 4.** Dependency between number of leading zero bits and wall clock execution time of the signature operation.

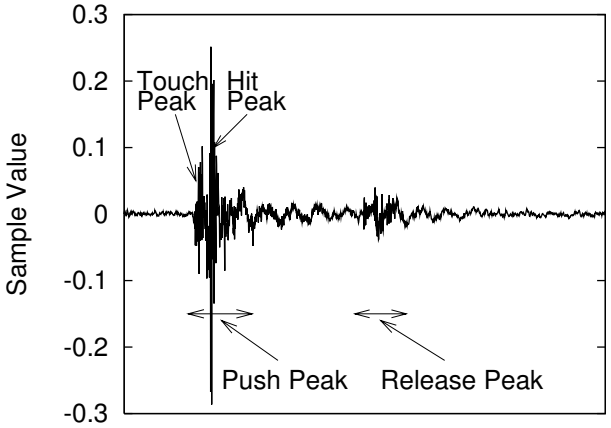
**Reference:** *Remote Timing Attacks are Still Possible*, Brumley et al., ESORICS 2011
http://packetstorm.linuxsecurity.com/papers/cryptography/timing-attacks.pdf

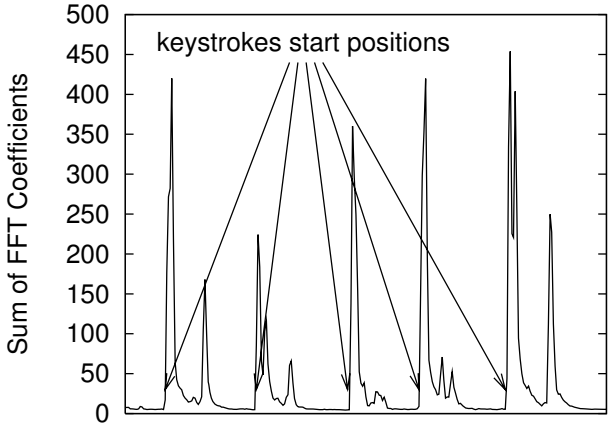
# Fault Injection Attacks

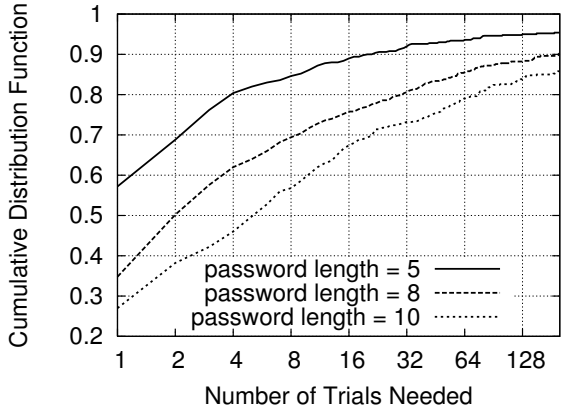


**Reference:** *Secure Application Programming in the Presence of Side Channel Attacks*, Whitteman et al., RSA Conference 2008
http://www.riscure.com/benzine/documents/Paper_Side_Channel_Patterns.pdf

# Audio Analysis

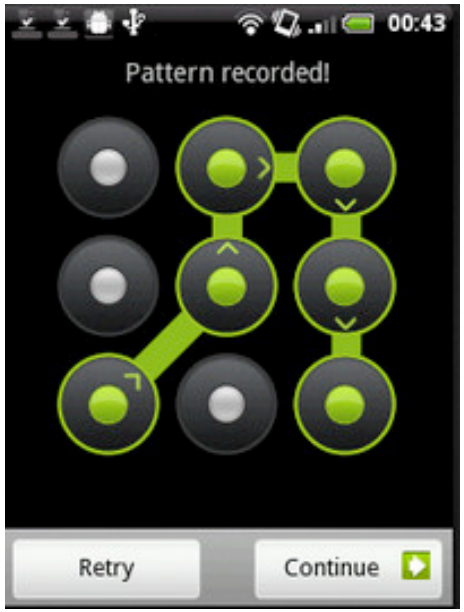


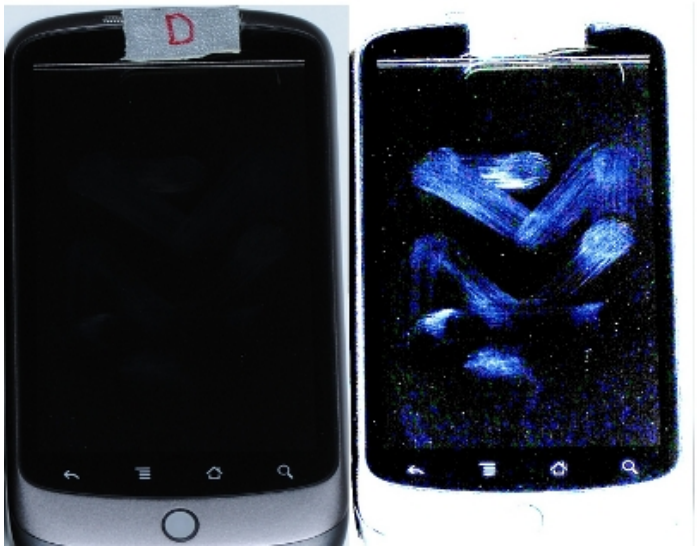




**Reference:** *Keyboard Acoustic Emanations Revisited*, Zhuang et al., TISSEC 2009
http://www.tygar.net/papers/Keyboard_Acoustic_Emanations_Revisited/ccs.pdf

# “Smudge Attacks”




**Reference:** *Smudge Attacks on Smartphone Touch Screens*, Aviv et al..
http://www.usenix.org/event/woot10/tech/full_papers/Aviv.pdf

# Video Attacks

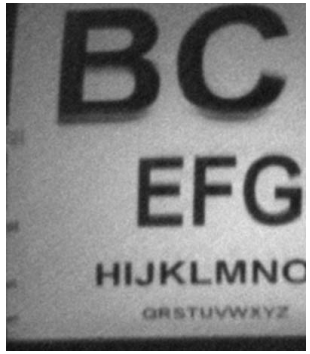

**Reference:** *ClearShot: Eavesdropping on Keyboard Input from Video*, Balzarotti et al., IEEE Conf on Security and Privacy 2008
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.207.4423&rep=rep1&type=pdf

**Reference:** *iSpy: Automatic Reconstruction of Typed Input from Compromising Reflections*, Raguram et al., ACM Computer and Communications Security 2011
http://wwwx.cs.unc.edu/~amw/resources/ispy.pdf

**Reference:** *MAPS: Machine-based Automatic Phone Surveillance*, Goswami et al., DCS Technical Report, UNC Chapel Hill, 2011
ftp://ftp.cs.unc.edu/pub/techreports/11-004.pdf

# Video Attacks



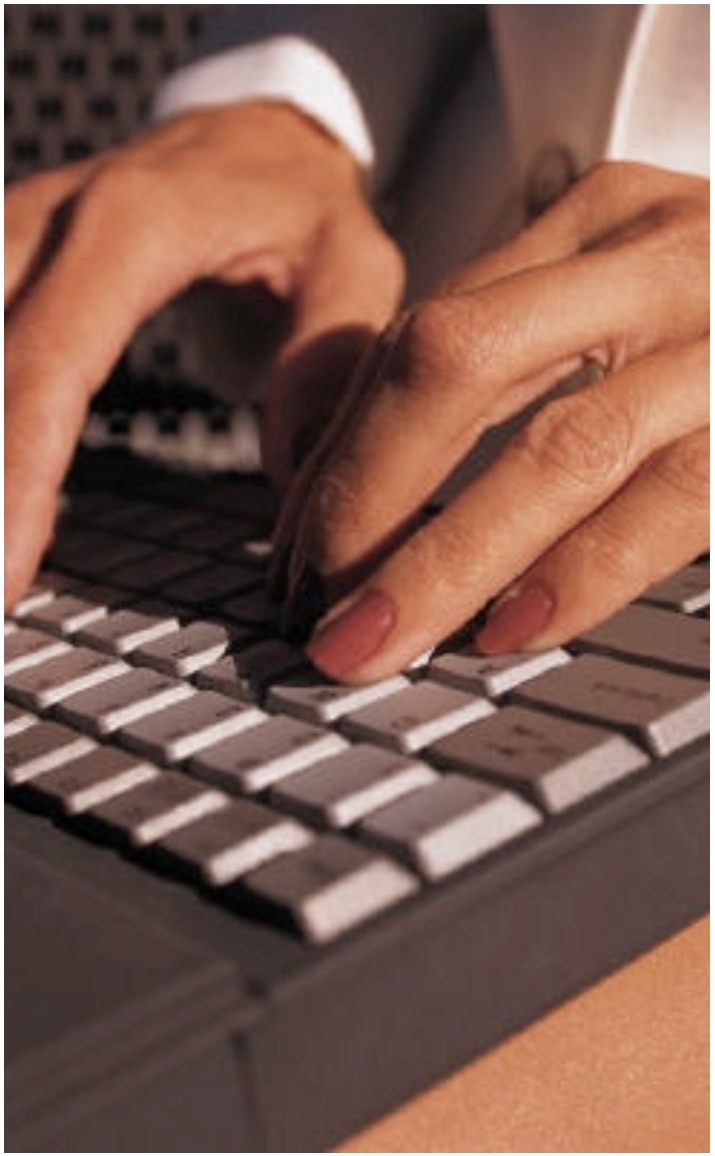



**Reference:** *Compromising Reflections – or – How to Read LCD Monitors Around the Corner*, Backes et al.., IEEE Symp. on Security & Privacy 2008
http://gauss.ececs.uc.edu/Courses/c653/extra/reflections.pdf

# Document Metadata

Additional “metadata” may be inadvertently included in documents:

- Document editing history
- Attempts to redact sensitive data
- Image EXIF headers
- etc..



# Certification Schemes

# Certification Schemes

As computer systems proliferated, so did systems claiming to increase security

- Organizations had a hard time deciding which security system to use for a certain job
- It would take too much time/effort to evaluate a system

The solution was to create a **certification scheme** that describes the security features of a system

- An independent third party evaluates a system and assigns a "rating" based on the system design and its capabilities
- The evaluation indicates how likely the system will be secure, but it does not say anything about whether the system has vulnerabilities

# Example: TCSEC (Orange Book)

**TCSEC/Orange book** was developed by the U.S. Government (1983-2005)

- Used for military and government applications to help agencies determine if the components they are using have appropriate security safeguards
- It was the first major security evaluation methodology in general use and remained the most popular for almost 16 years
- There were 7 levels of security systems: D, C1, C2, B1, B2, B3 and A1 (from least to most secure)

Certification Levels

- **D: Minimal protection**
  - Anything that fails to meet even the lowest C1 designation

# Example: TCSEC (Orange Book)

**C: Discretionary Protection**

- C1: Discretionary Security Protection (UNIX/Linux)
  - Implements DAC to enforce limits on user access to objects
- C2: Controlled Access Protection (Windows)
  - Audit trails, object reuse protection

**B: Mandatory Protection**

- B1: Label Security Protection
  - MAC system (i.e. BLP), labels on select subjects/objects
- B2: Structured Protection
  - labels on all subjects/objects, covert channel analysis
- B3: Security Domains
  - Trusted path, full reference monitor requirements, constraints on development technique

**A: Verified Protection**

- A1: The same as B3 + formal verification of the system

# Example: Common Criteria

**Common Criteria** was developed in 1998 by an international group of countries

- Countries include Canada, USA, UK, France, Germany, Netherlands, etc.
- Adopted by US government in 2005 (replaces TCSEC)
- Allows users to **specify** their security requirements
- Allows vendors to **implement** and/or make claims about the security attributes of their products
- Allows testing laboratories to **evaluate** the products to determine if they actually meet the claims

- Uses levels called Evaluation Assurance Levels
  - There are 7 levels EAL1 (lowest) to EAL7 (highest)
  - Each level corresponds roughly to an Orange Book classification

# Evaluation Assurance Levels 1 – 4

**EAL 1:** Functionally Tested
- Review of functional and interface specifications
- Some independent testing

**EAL 2:** Structurally Tested
- Analysis of security functions, including high-level design
- Independent testing, review of developer testing

**EAL 3:** Methodically Tested and Checked
- Development environment controls, configuration management

**EAL 4:** Methodically Designed, Tested, Reviewed
- Informal spec of security policy, independent testing

# Evaluation Assurance Levels 5 – 7

**EAL 5:** Semiformally Designed and Tested

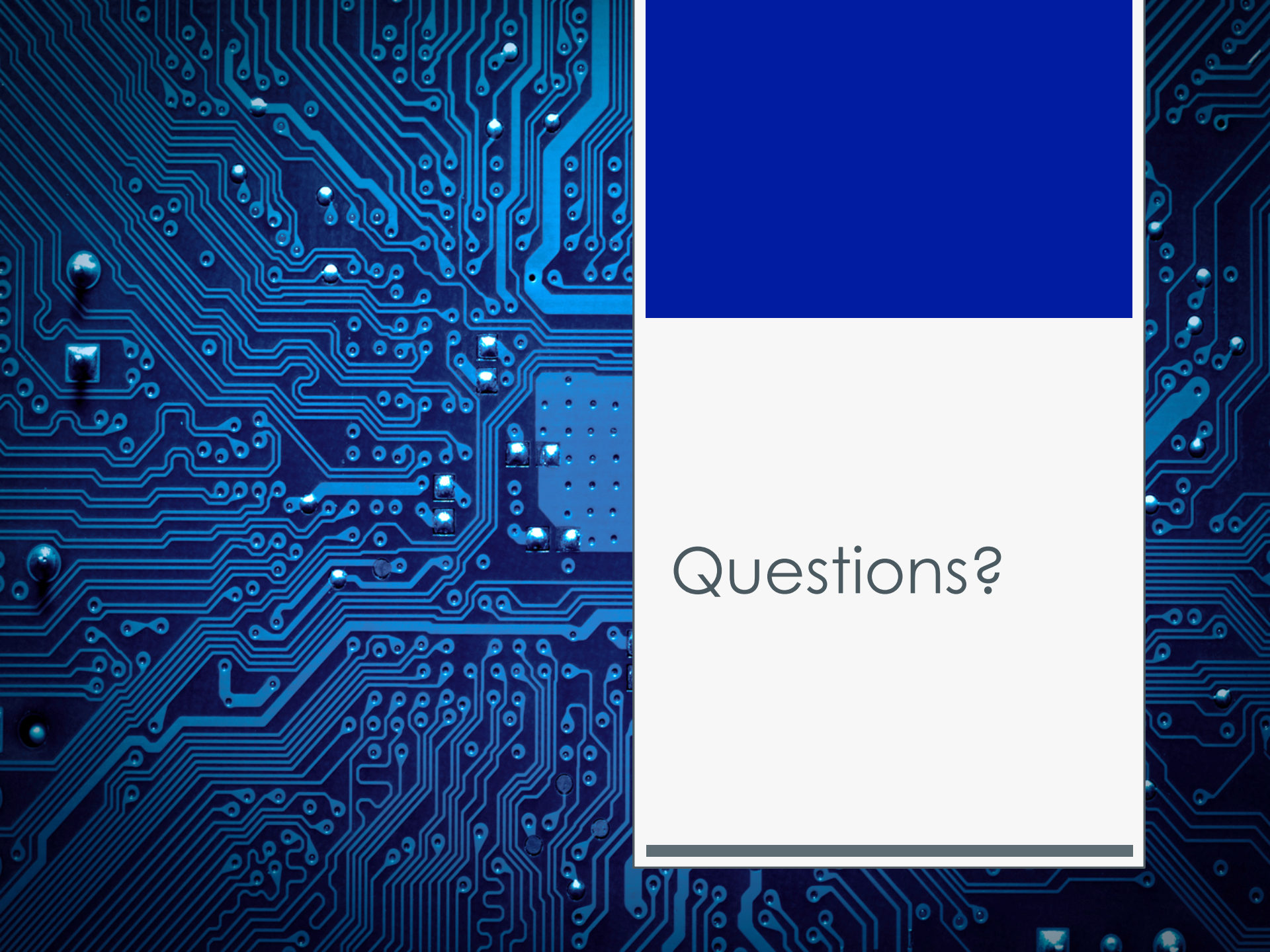
- Formal model, modular design
- Vulnerability search, covert channel analysis

**EAL 6:** Semiformally Verified Design and Tested

- Structured development process

**EAL 7:** Formally Verified Design and Tested

- Formal presentation of functional specification
- Product or system design must be simple
- Independent confirmation of developer tests


Questions?