Building Robust Security Solutions Using Layering And Independence

Fred Roeper **Neal Ziring**

Information Assurance Directorate
National Security Agency

Forward. Thinking.

Session ID: STAR-401

Session Classification: Intermediate

RSA CONFERENCE 2012

Introduction

Purpose:

- Provide information about layered design techniques for secure systems, to allow you to apply this approach to your own systems.

Outline:

- Introduction to layered design and NSA's Commercial Solutions for Classified initiative
- Principles of independence
- Case studies
- Evaluating independence
- Applying layered design

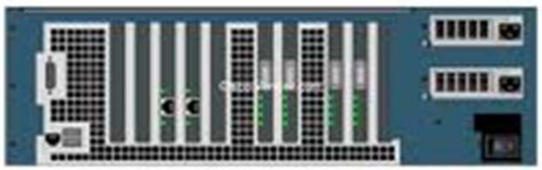
Introduction to Layered Design and CSfC

RSACONFERENCE2012

GOTS versus COTS

Traditionally, the US government has used government designed and certified devices to protect its most sensitive data.

- Government Devices (GOTS)
  - Purpose-built for security
  - Strict design and implementation criteria
  - Long, exhaustive security evaluation



- Commercial Devices (COTS)
  - Provide a balance of security and features
  - Quick to market, flexible

GOTS versus COTS

- **GOTS**:
  - Assurance: high
  - Lifecycle costs: high
  - Development: slow
  - Gov't control: high
- **COTS**:
  - Assurance: varies
  - Lifecycle costs: lower
  - Development: quick
  - Gov't control: low

Q. ***How can we enjoy the agility and flexibility of commercial devices, with assurance sufficient to protect the most sensitive national security information?***

Solution

- Enumerate all the individual security requirements needed to achieve overall assurance objectives (confidentiality, integrity, etc.)
- For each security requirement:

  → Provide *multiple* mechanisms that satisfy the requirement,

  ..such that each mechanism is *sufficient* should another get compromised,

  ..such that the mechanisms are *independent:* vulnerability or compromise of one does not imply compromise of another.

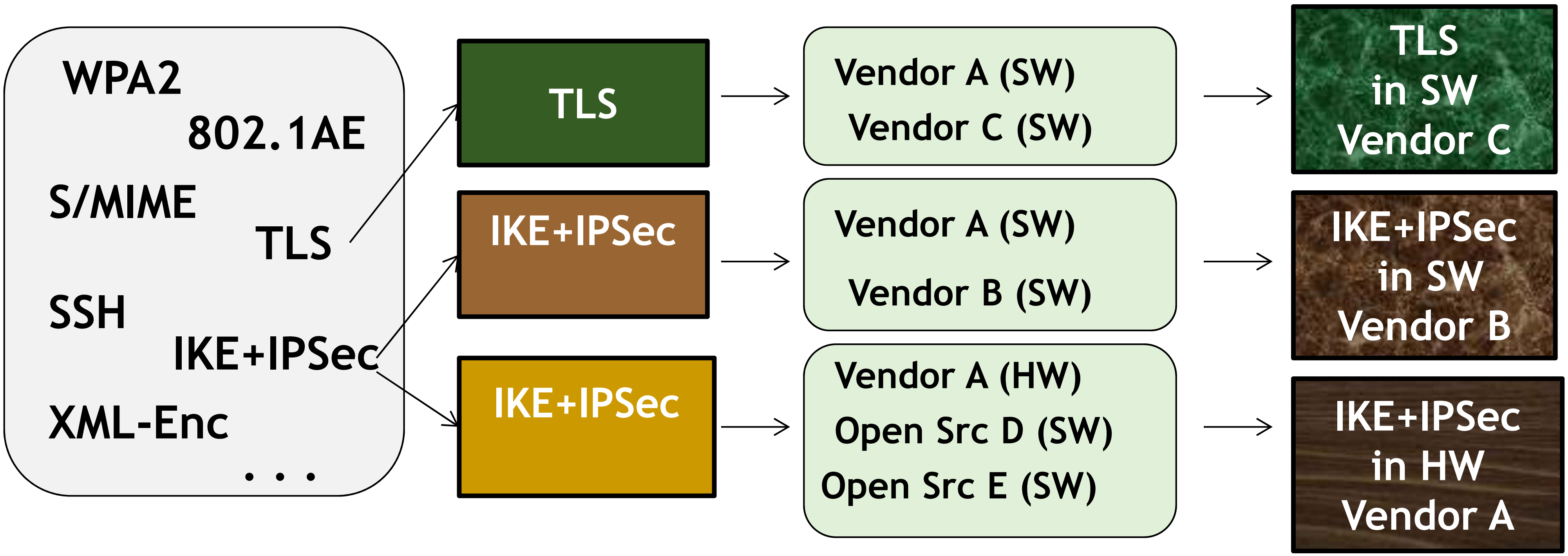
- Supplement with detection mechanisms that can monitor health of the primary mechanisms.
- **This approach can be used by anyone who needs greater assurance, not just government.**

Security through Composition

Example:

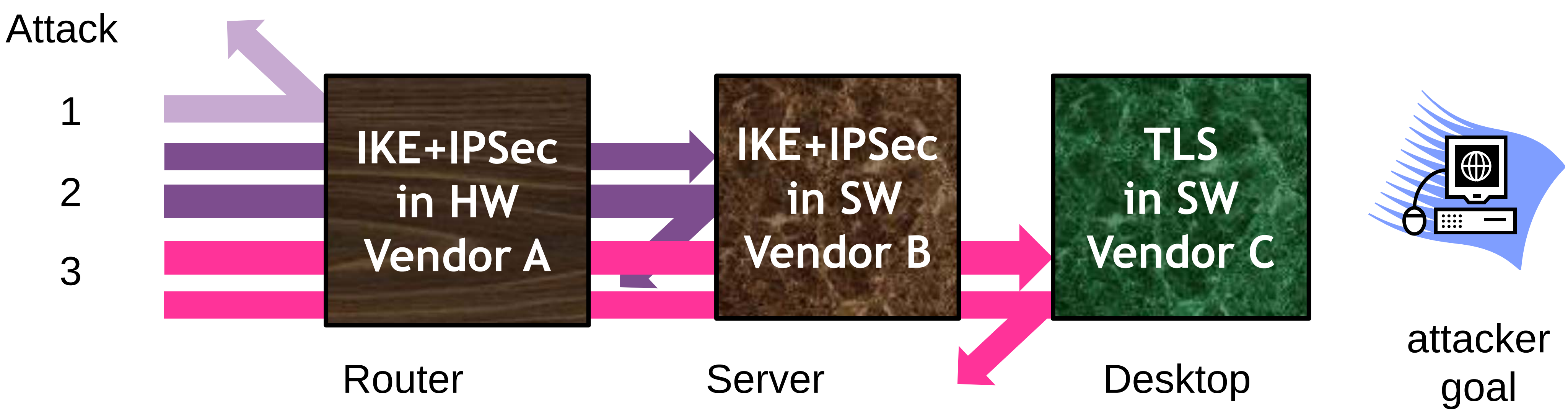
- IA requirement: confidentiality - prevent unauthorized access to information in transit over an untrusted network

Select standards → Compose → Select implementation → Implement



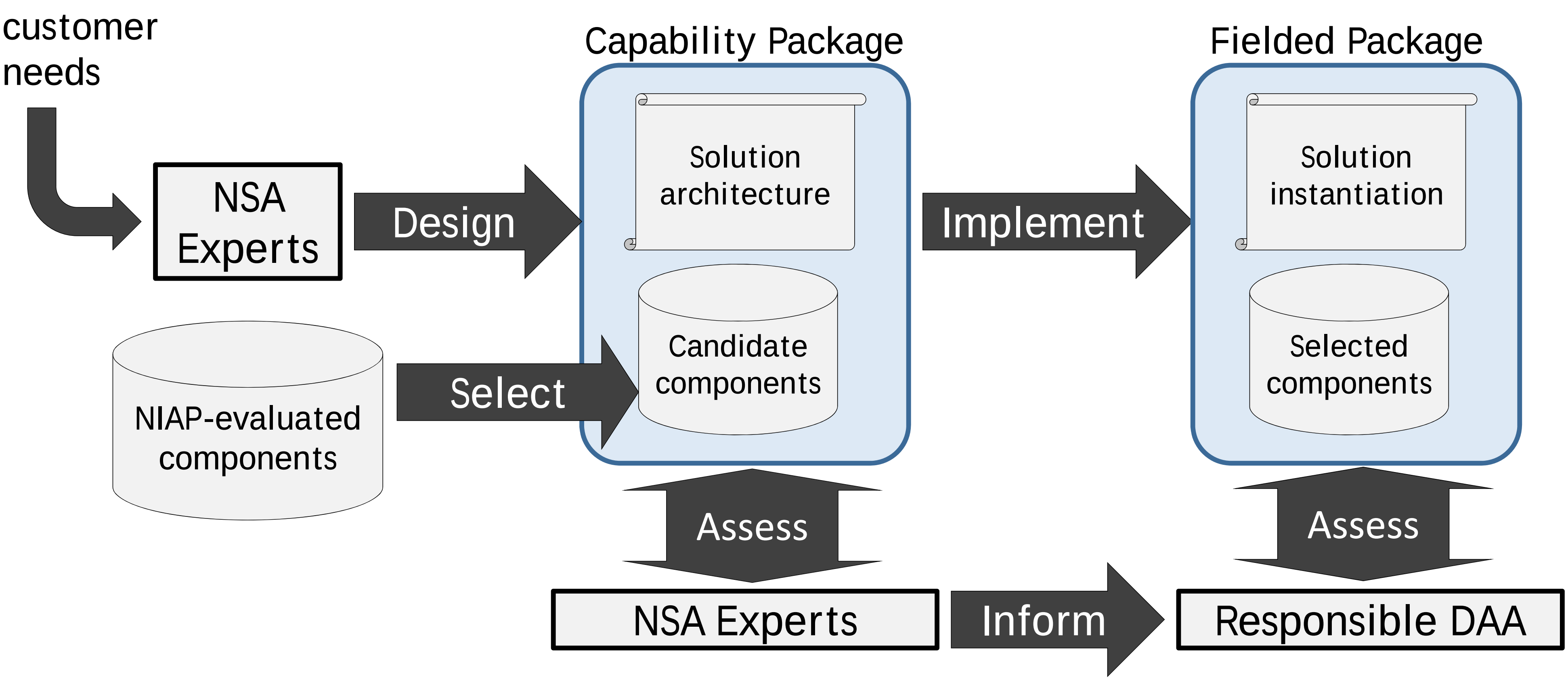
Security through Composition

- Layered security is most effective when the layers exhibit <u>independence</u>.
- Means to achieve independence:
  - Different algorithms, processors, suppliers, software, protocols, platform, staffing, operations, configuration, . . .



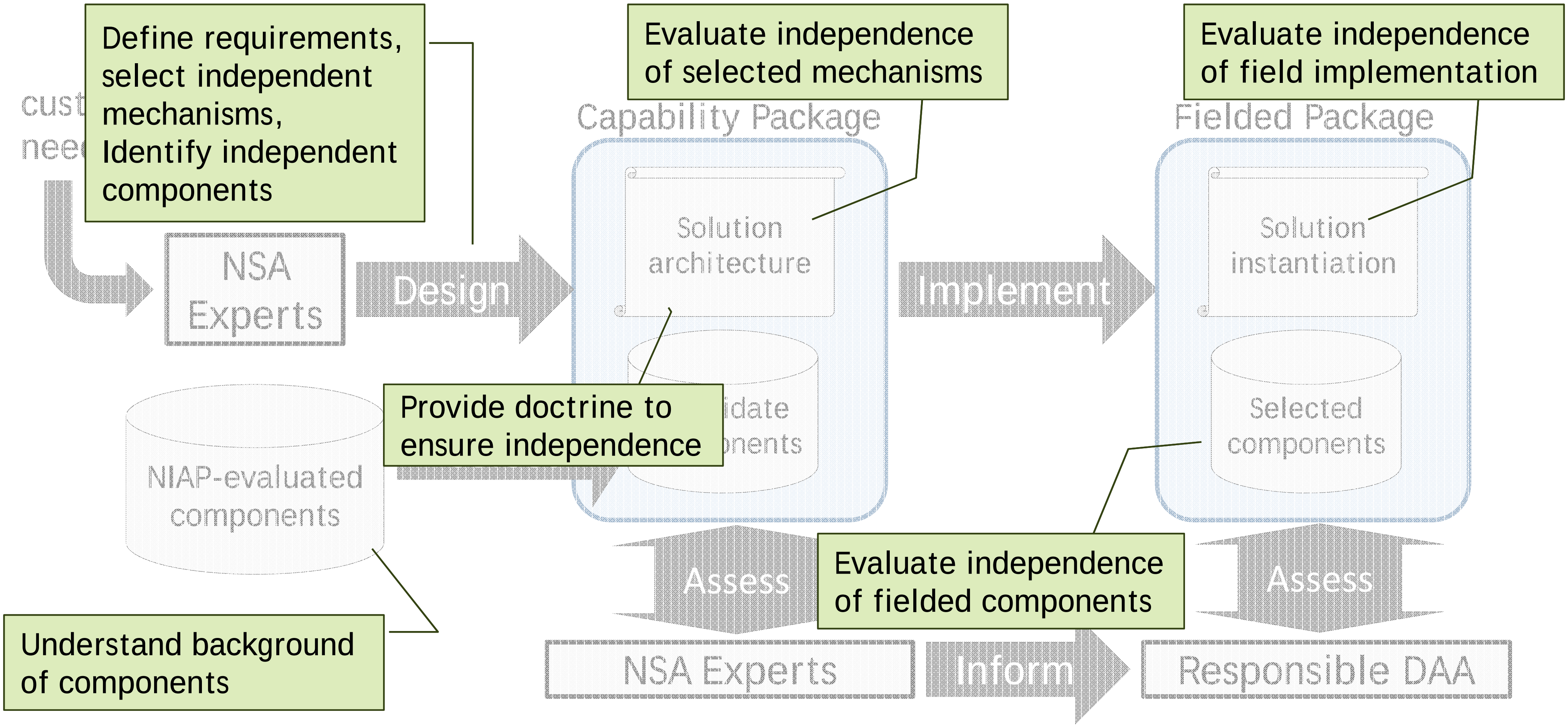
Example: NSA Commercial Solutions Strategy

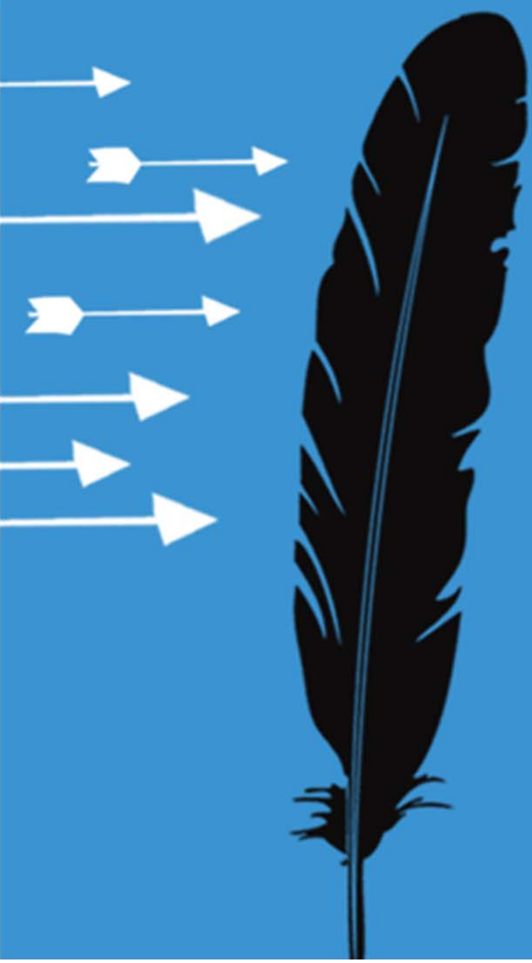
- The NSA Commercial Solutions for Classified (CSfC) process uses composition to increase assurance.



Example: NSA Commercial Solutions Strategy

- The NSA Commercial Solutions for Classified (CSfC) process uses composition to increase assurance.





Principles of Independence

Types of Independence

- Coupling is the opposite of independence.
  - Coupling is usually based on common element(s) or shared pedigree.
  - Coupling can be based on any part of a mechanism or component's lifecycle: foundation, concept, design, implementation, deployment, etc.
  - Coupling in multiple parts of the lifecycle reduces independence.
- Examples:

Form of Independence	Common Element	Parts of Lifecycles
Algorithmic	Same cryptographic algorithm	Concept, design
Credential	Same provider of keys, secrets	Deployment, operation
Codebase	Same underlying source code	Implementation
Administrator	Same privileged administrators	Operation

Forms of coupling where the common element is less assured require greater attention to independence.

Degree of Independence: Spectrum

Independence between layered security mechanisms varies depending on:

- Design (e.g., algorithms, protocols, architecture)
- Implementation (e.g., libraries, development tools, platforms, etc.)
- Implementer (e.g, coders, testers, suppliers, integrators, etc.)
- Operation (e.g., installers, administrators, auditors, etc.)
- . . .

Fully coupled
(no independence)

Fully independent

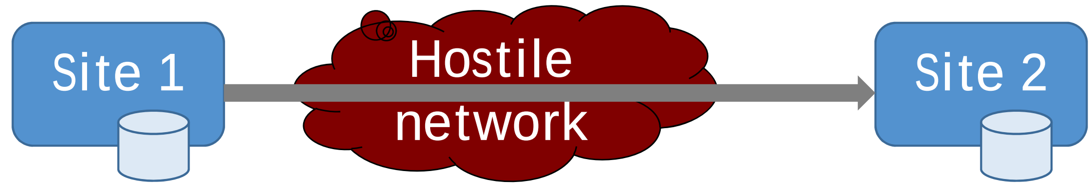
Very Low Independence	*Medium Independence*	**High Independence**
Brand A Hardware Encryptor / Brand A Hardware Encryptor	Brand A Hardware Encryptor / Brand C Hardware Encryptor	Brand B Software Encryptor / Brand A Hardware Encryptor
Same algorithm	*Same algorithm*	*Same algorithm*
Same protocol	*Same protocol*	**Different protocols**
Same embedding	*Same embedding*	**Different embedding**
Same platform	**Different codebases**	**Different platforms**
Same codebase	**Different platforms**	**Different codebases**
Same key source	**Different key sources**	**Different key sources**
Same administrators	*Same administrators*	**Different administrators**

Types of Independence: Prevention v. Detection

Independence can be gained through differences in:

- **Preventative controls** –
  these directly enforce/satisfy the security requirement
  - Encryption
  - Access control
  - Port filtering
  - User authentication & authorization
  - *Etc.*
- **Detective controls** –
  these detect violation of the requirement so it can be mediated
  - Auditing and log inspection
  - Configuration management
  - Intrusion detection
  - Flow monitoring
  - *Etc.*

Audience Exercise



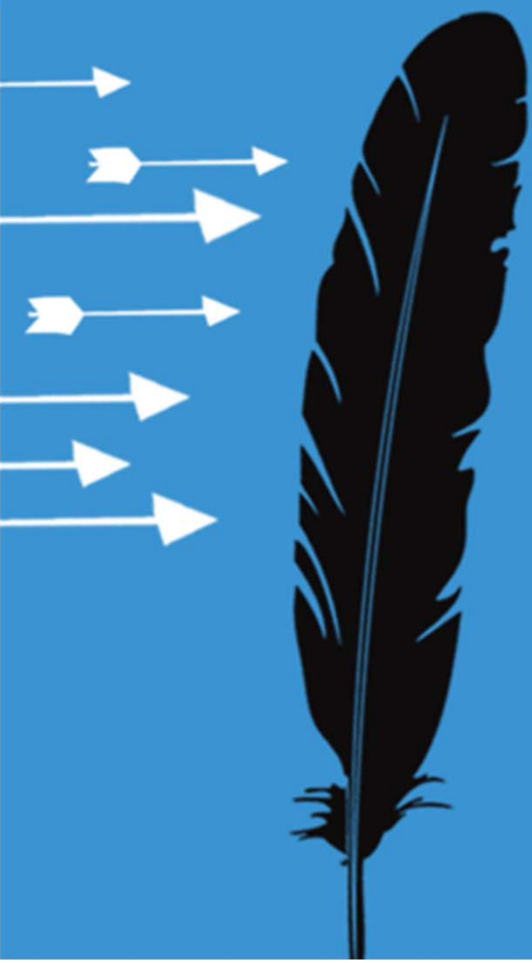
- Security requirement: **file integrity** (file must not change in transit)
- Which design has more independence between **A** and **B**?

Design 1:

- Mechanism **A**
  sign file in CMS format, using PKI cert from enterprise CA, software from vendor X
- Mechanism **B**
  HTTPS (TLS) transfer, using PKI cert from enterprise CA, software from vendor Y

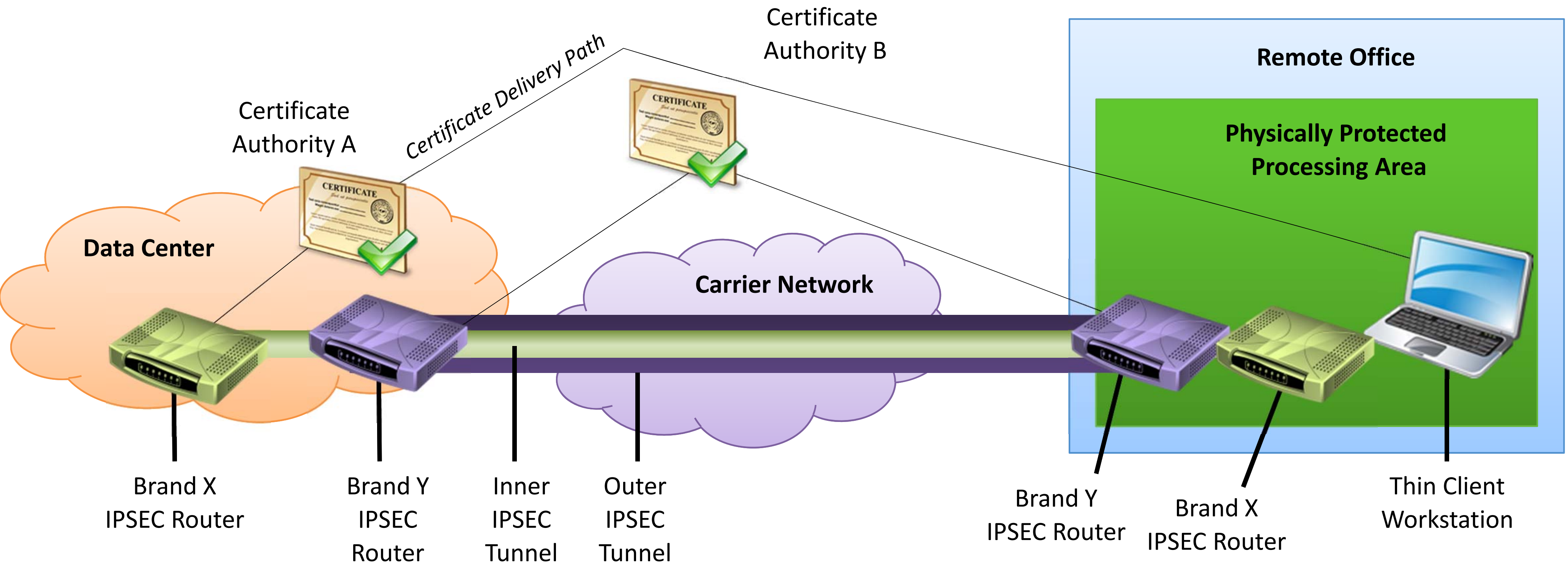
Design 2:

- Mechanism **A**
  sign file in XML format, using PKI cert from enterprise CA, software from vendor X
- Mechanism **B**
  IPSec VPN between sites, using pre-placed key, hardware from vendor Z

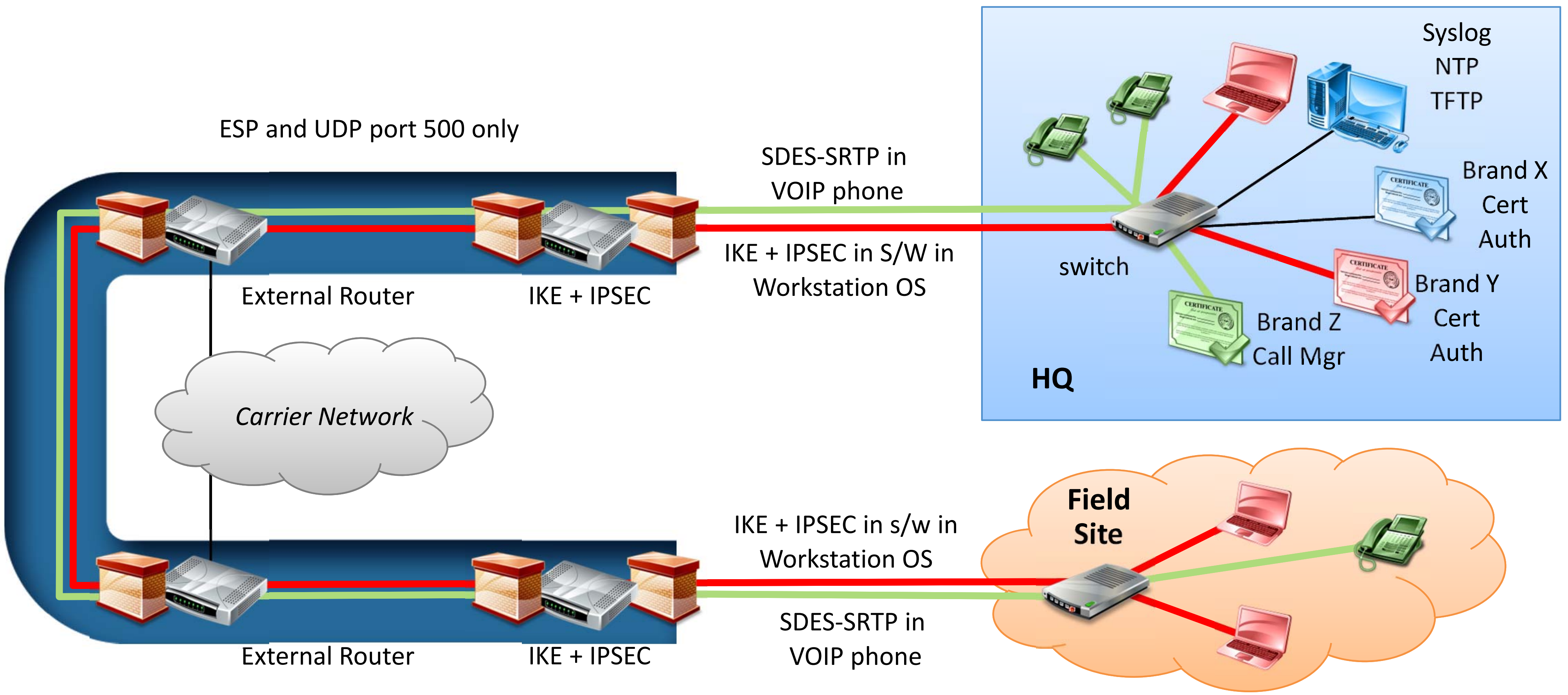


Case Studies: Layered System Designs

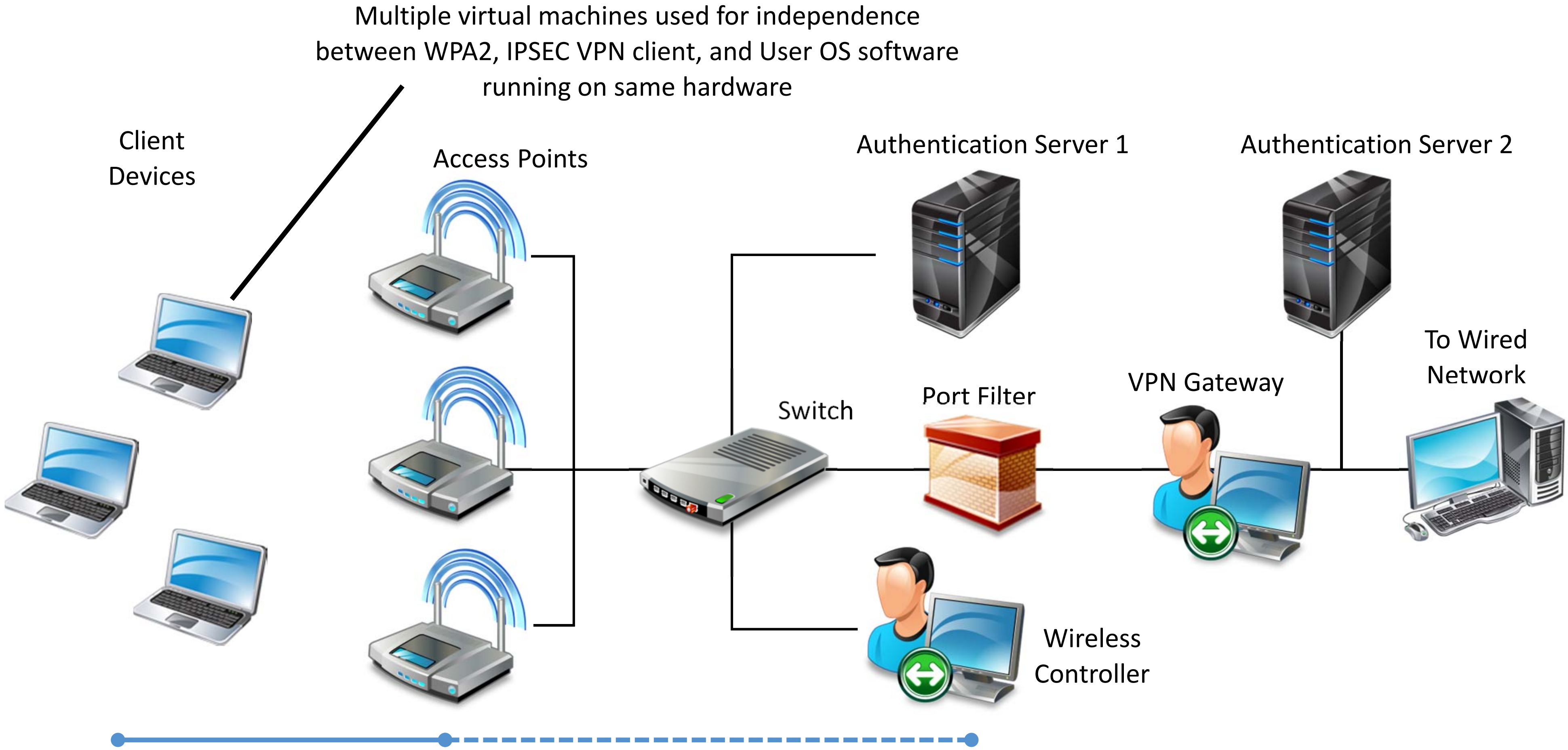
Remote Office VPN



Field Site VPN

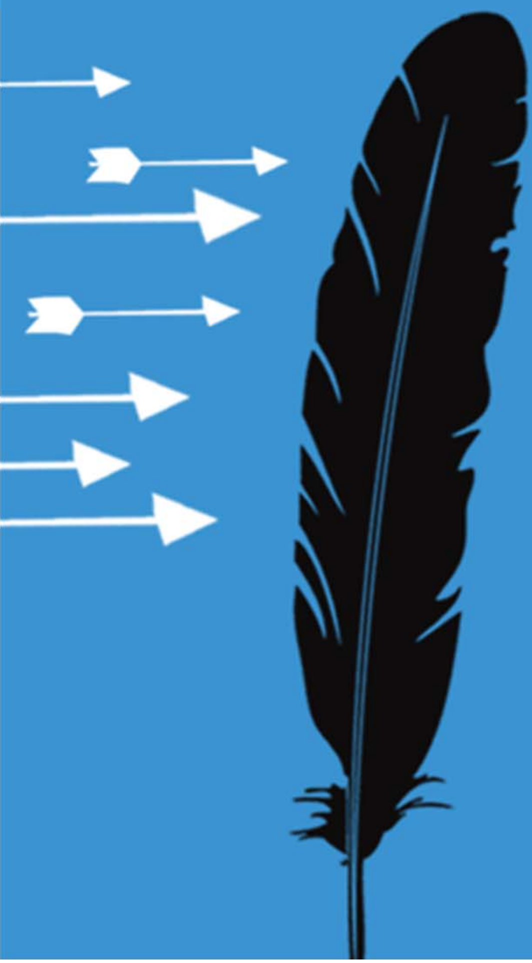


Secure WiFi



WPA2 Enterprise using 128-bit AES-CCMP FIPS 140-2 validated encryption with EAP-TLS authentication passing X.509 machine certificates

IPsec VPN using 128-bit AES CBC FIPS 140-2 validated Encryption with X.509 machine certificates



Evaluating Independence

Evaluating Security of Layered Independent Protection

- Protection mechanisms are selected to be sufficient to protect on their own, **if effective**
  - So we only need ***at least one effective*** protection mechanism in order to be secure
- But we **don't have full assurance** in mechanisms
  - Let A_i be the % **Assurance of Mechanism** i
- And **mechanisms aren't always 100% independent** of each other
  - Let I_i be the % of **Independence of Mechanism** i from all other layered mechanisms
- **Layered Assurance =** $\mathbf{1 - \prod_{i=1 \ldots n} (1 - I_i A_i)}$
  - Or Layered Assurance = $1 – (1-I_1A_1)(1-I_2A_2) \ldots (1-I_nA_n)$
  - Where n is the number of layered mechanisms
  - If we assume 100% independence between all mechanisms, then
  - 2 Layer Example: 1 – (1-75%)(1-80%) = 95% layered assurance
  - 3 Layer Example: 1 – (1-60%) (1-75%) (1-80%) = 98% layered assurance
  - 4 Layer Example: 1 – (1-50%) (1-60%) (1-75%) (1-80%) = 99% layered assurance

Evaluating Security of Layered Independent Protection: Method

- How to determine **Assurance of Mechanism**?
  - Process integrity
  - Compliance with standards
  - Testing
  - Trust in developer
  - Trust in suppliers of subsystems/components
  - Historical record of vulnerability

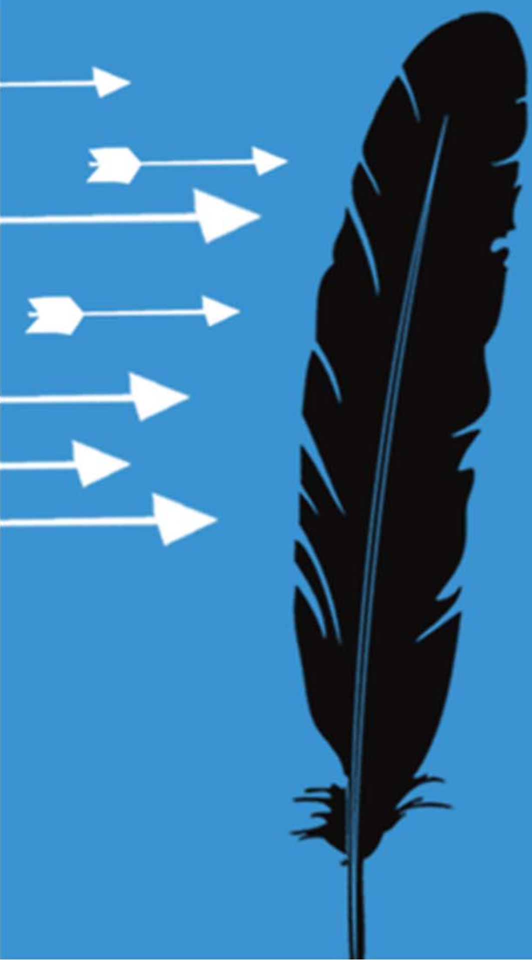
Evaluating Security of Layered Independent Protection: Method

- How to determine **Independence of Mechanism**?
  - A mechanism is independent to the degree that its factors of independence are different from those same factors in all other layered mechanisms
- I = (# of factors independent of all other layers / # of factors)

Factor of Independence	Mechanism A	Mechanism B	Mechanism C
algorithm	*Algorithm 1*	*Algorithm 1*	Algorithm 2 ✓
protocol	Protocol 1✓	Protocol 2 ✓	Protocol 3 ✓
embedding	Embedding 1 ✓	Embedding 2 ✓	Embedding 3 ✓
platform	*Platform 1*	Platform 2 ✓	*Platform 1*
codebase	Codebase 1 ✓	Codebase 2 ✓	Codebase 3 ✓
key source	Key source 1 ✓	Key source 2 ✓	Key source 3 ✓
administrators	*Admins 1*	*Admins 1*	*Admins 1*
supplier	Supplier 1 ✓	Supplier 2 ✓	Supplier 3 ✓
Degree of Independence	**62.5%**	**75%**	**75%**

Evaluating Security of Layered Independent Protection: Examples

- Three layer example using Assurance of Mechanism and Degree of Independence values from previous examples:
- Layered Assurance = $1 – (1-I_1A_1)(1-I_2A_2)(1-I_3A_3)$
- Mechanism 1: $I_1 = 62.5\%$, $A_1 = 60\%$,
- Mechanism 1: $I_2 = 75\%$, $A_2 = 75\%$
- Mechanism 1: $I_3 = 75\%$, $A_3 = 80\%$
- Layered Assurance = $1 – (1-0.375)(1-0.5625)(1-0.6)$
- Layered Assurance = **89.0625%**
  - Layered Assurance was 98% with I_1, I_2, $I_3 = 100\%$



Wrap-up

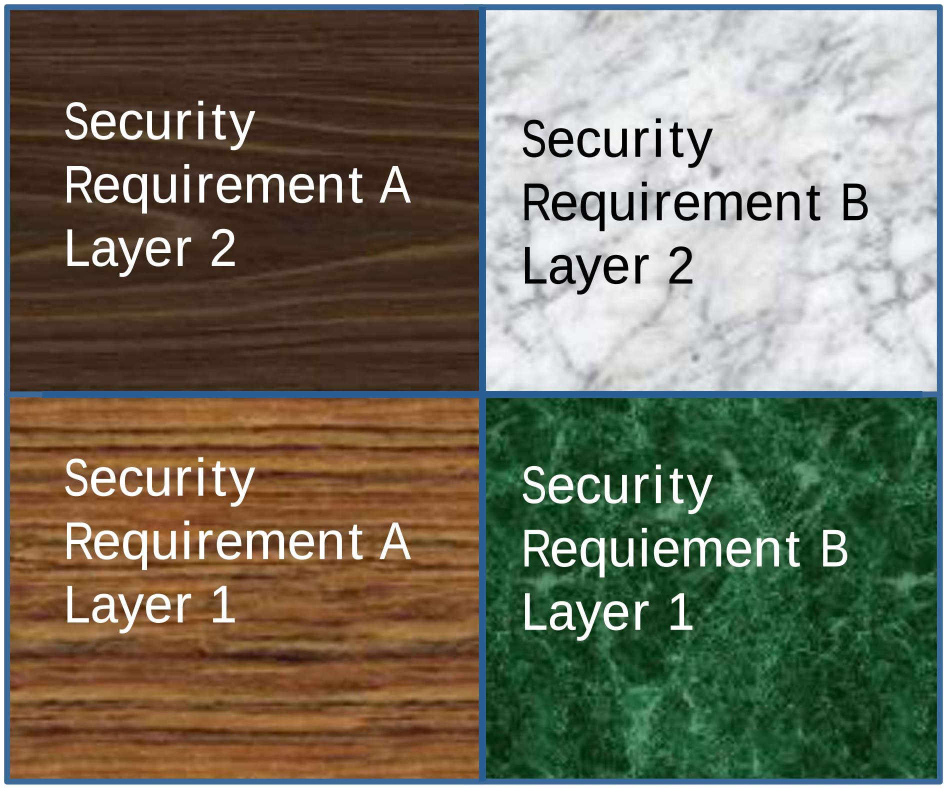
Application

Apply layered design for your secure system projects:

1. Select set of security requirements
   (defense in depth – cover all forms of IA needed for your information & systems)
2. For each requirement:
   - Identify candidate mechanisms
   - Assess independence of mechanisms
   - Select mechanisms and compose
3. Implement composite design
   - Select specific implementations for each mechanism
   - Assess independence of implementations
   - Assess assurance of implementations
4. Deploy composite design
   - Establish independence in operation and management

Application

- Remember this when designing your next secure system:



Independent Layers of Defense

Defense In Depth

Summary

- Composition of security mechanisms (layering) can be used to gain assurance.
- Independence of mechanisms is a primary criterion for judging members of a composition.
  - Many different forms of independence across component lifecycle
  - Multiple forms apply to any candidate composition.
- Layered Assurance can be analyzed using probability
  - **Layered Assurance = $1 - (1 - I_1 A_1)(1 - I_2 A_2) \dots (1 - I_n A_n)$**
- Apply this method to create more assured systems:
  - Select IA requirements
  - Identify multiple mechanisms/components for each requirement
  - Assess the independence of mechanisms for each requirement
  - Assess the assurance of mechanisms for each requirement
  - Calculate the layered assurance for each requirement
  - Evaluate whether resulting assurance meets Information Assurance objectives

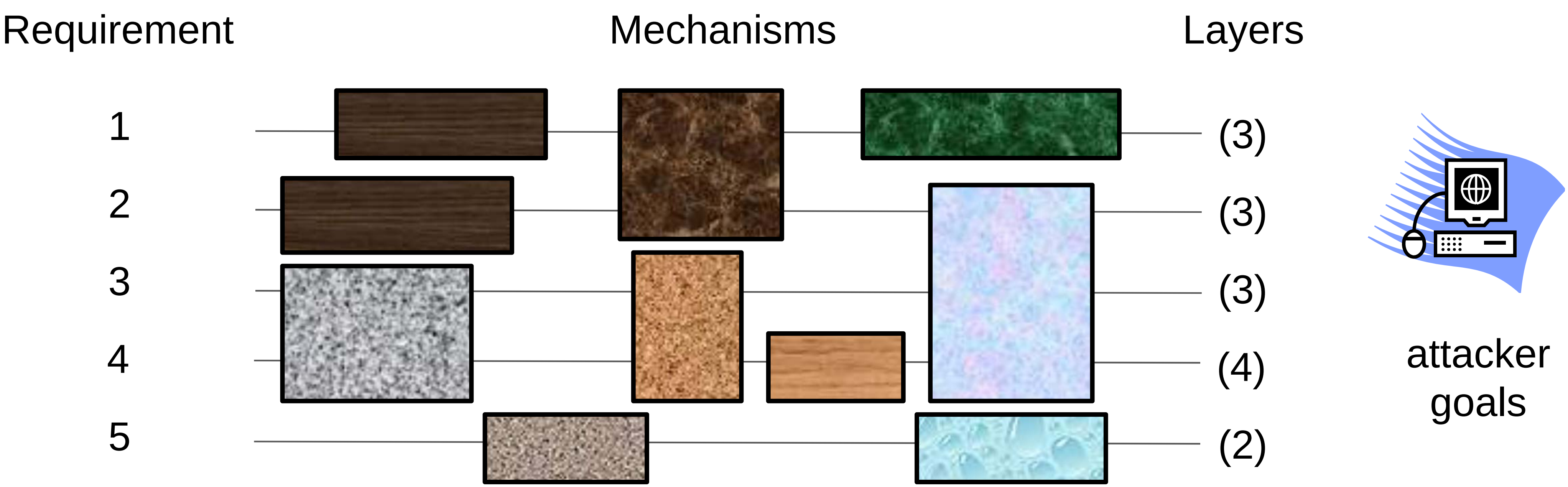
Future Work

- Areas for further research and refinement:
  - More rigorous definition of various forms of independence
  - Incorporation of data necessary for evaluating independence into security certification regimes (e.g., NIAP)
  - Practical design rules for composition of security mechanisms
  - Improved test criteria and testing tools for composite systems
  - Formal or scientific basis for measuring confidence
  - Operational mechanisms for monitoring independence in fielded systems

Backup Slides

Security through Composition

- Each security requirement has multiple mechanisms supporting.
- Some mechanisms may support multiple requirements.
- Each set has adequate independence.



Types of Independence (Longer List)

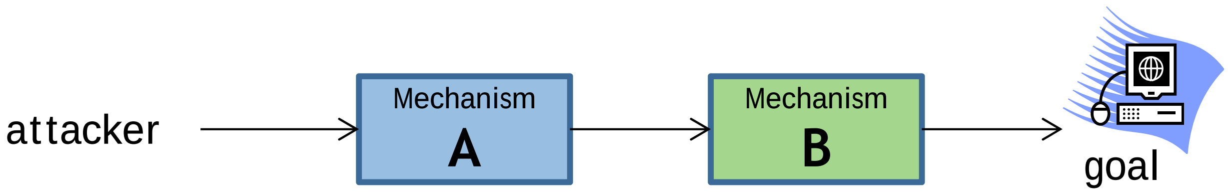
- Math foundation
- Algorithm
- Standards body
- Protocol
- Code library (crypto library)
- Credential
- Entropy
- Embedding (HW,SW,etc.)
- Language
- Codebase

- Dev. Tools
- Dev. environment
- Operating system
- Developer
- Supplier
- Installer
- Network
- Audit
- Location/Physical
- Administrator
- Configuration

- Operator
- Oversight
- Maintenance
- Disposal
- Backup
- Power & Cooling
- Management plane
- Control plane
- Data plane
- National origin

Degree of Independence: Spectrum

- Independence between layered components supporting the same security requirement varies along a spectrum based on how successful compromise of one affects the work needed (or chance of success) for compromising another.



Fully coupled

Compromise of A reduces the work to compromise B to zero.

Fully Independent

Compromise of A provides no advantage for compromising B.

Anti-coupled

Compromise of A makes compromise of B infeasible