E-Commerce Security

Raising Awareness of Issues by Adapting the NIST IT Security Services Model to E-Business Systems

Robert L. Probert, Victor Sawma[1]

School of Information Technology and Engineering

University of Ottawa

March 6, 2003

[1] (Second Author on leave from Notre Dame, Beirut)

Abstract

More and more government service providers and their suppliers are adopting the Electronic Business Models (B2C and B2B). Security features are key to reliability and trustworthiness. To help teach about critical security parameters for (IT) systems, the Common Criteria Redbook(1999) is an international standard for security. Very recently, (NIST) released the "Underlying Technical Models for IT Security" document. This explains and documents security features and known security attacks on IT systems.

As e-commerce systems are IT systems, we have adapted this to e-commerce. We select certain security features as requirements. We know most of the attacks against such features. Then, we include effective countermeasures in the design, even before coding. This avoids costly security breaches as well as expensive redesign and redevelopment.

In this presentation, we describe this methodology for deriving countermeasures for EC attacks based on the NIST security model for IS. The result of applying our methodology to each security feature in the NIST security services model is a countermeasures design model that is directly useful for ensuring secure EC systems.

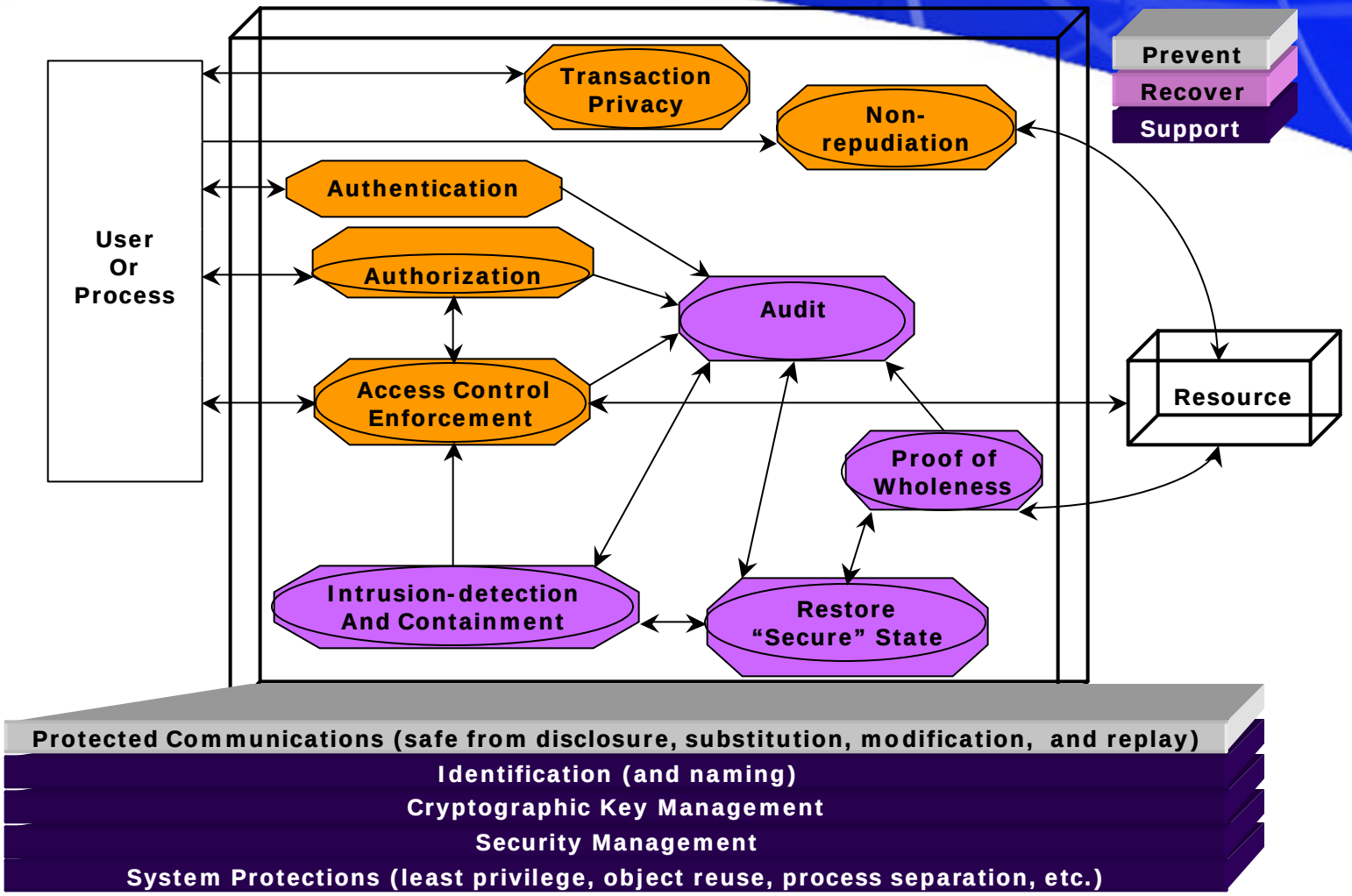
NOTE: The authors are very interested in receiving suggestions for the best means of teaching this design approach to systems analysts, designers and developers.

Outline of Presentation

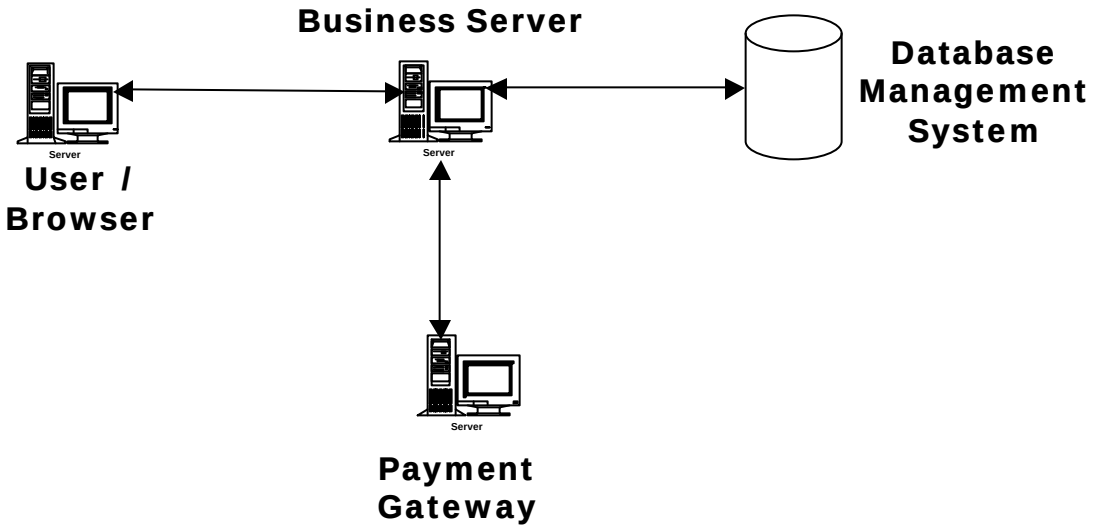
- Introduction and Background
- Current Industrial Practices & Motivation
- The Challenge
- Related Research
- Phase I – Derive Countermeasures (Models)
- Phase 2: Instantiate and Integrate the derived models into existing e-commerce system design
- Discussions and Conclusions
- Future Needs: **** Educational Strategy ****

Introduction and Background

- **Security Features and Requirements**
  - Legitimate user requirements support security features
  - Malicious user requirements "break / disable" security features
- **Common Criteria Redbook**
  - Legitimate user requirements
  - Evaluate IT systems
  - Requires system implementation
- **NIST Underlying Technical Models for IT Systems**
  - Legitimate user requirements
  - Security Services Model
  - Design IT systems
  - Must be specialized for use in EC systems
- **Open Source Security Testing Methodology Manual (OSSTMM)**
  - Malicious user requirements
  - Test an Internet system implementation
  - Cannot be used directly in designing EC systems

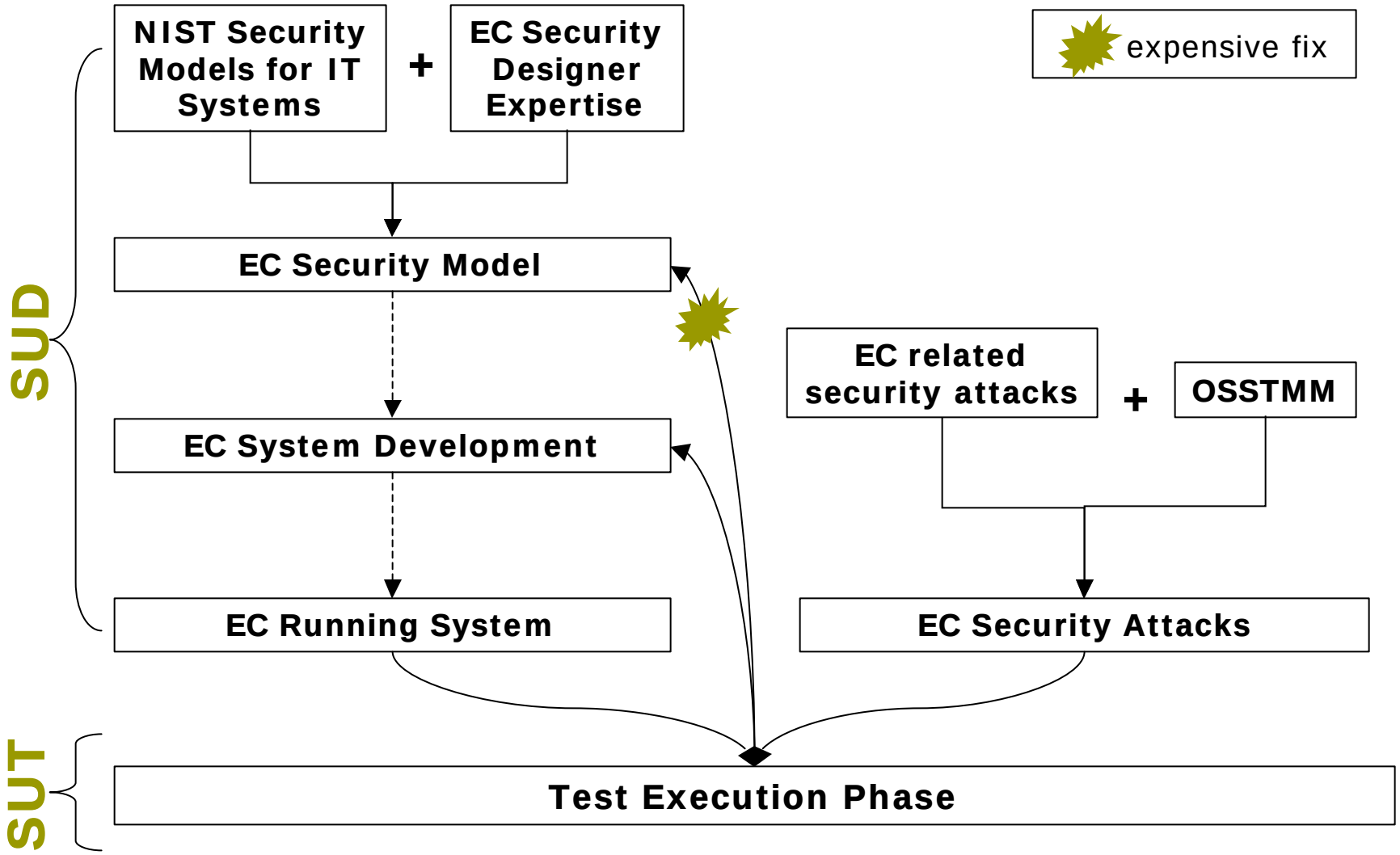


The NIST Underlying Technical Security Services Model for IT Systems



A Generic E-Commerce System Architecture

Current Industrial Practices & Motivation



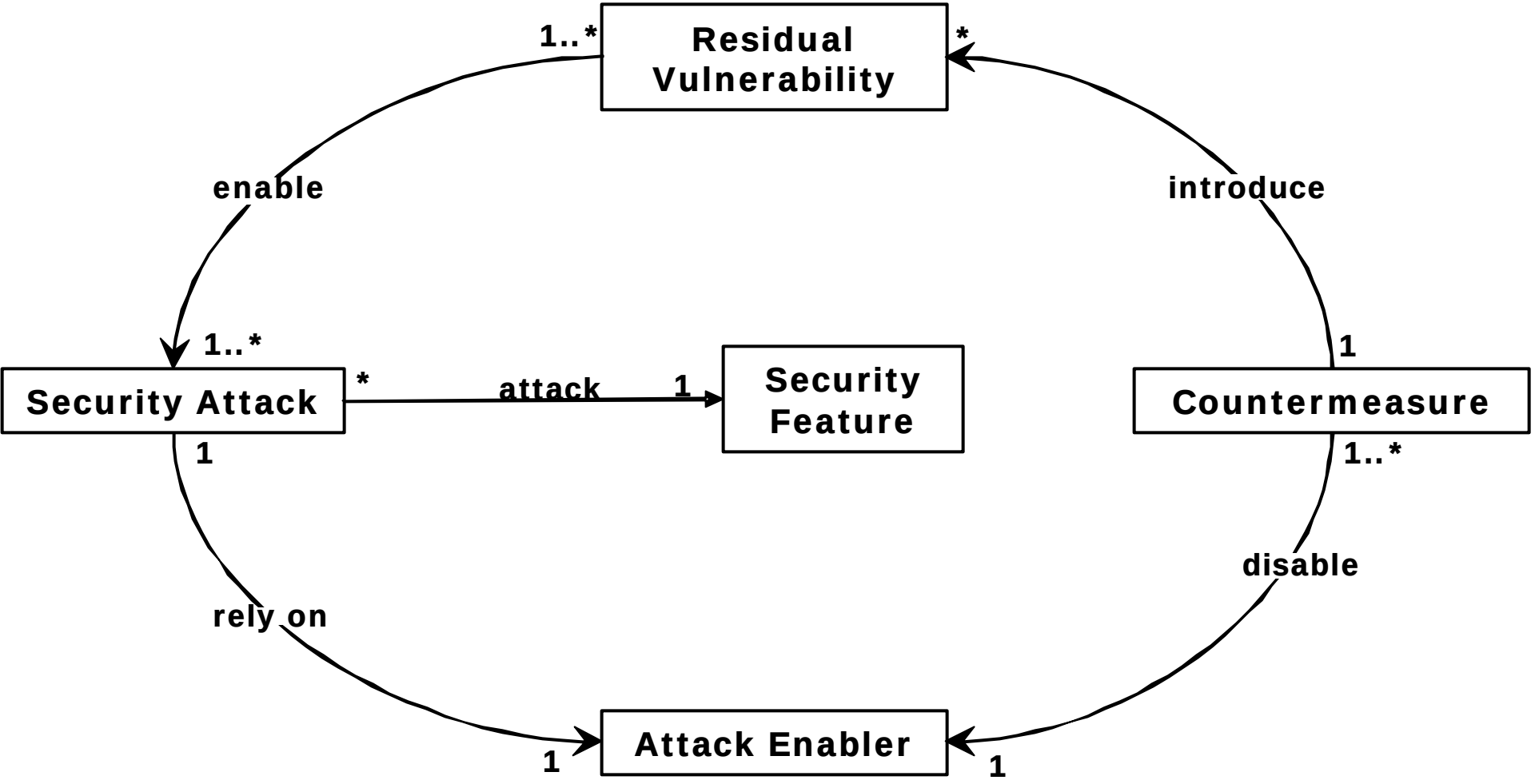
Problem Statement

How can appropriate security features, attributes, safeguards, and countermeasures be "**designed in**" to an **e-commerce system** under development, rather than waiting until the system is implemented and conducting a costly "**test and rework**" campaign?

Related Research

- Attack trees [Schneier]
  - **Attack-Goal centered**
  - List all possible attacks leading to goal
- Attack nets [McDermott]
  - Same as attack trees: Petri Nets instead of attack graphs
- Fault tree analysis [Fenelon]
  - **Fault/Problem centered**
  - List all possible causes for failure
  - Might help in component failure attacks
- **Our approach**
  - **Security-feature centered**
  - List all possible security attacks
  - How does our approach differ?
    - **There is a standard for all security features**
    - Research vs Expertise

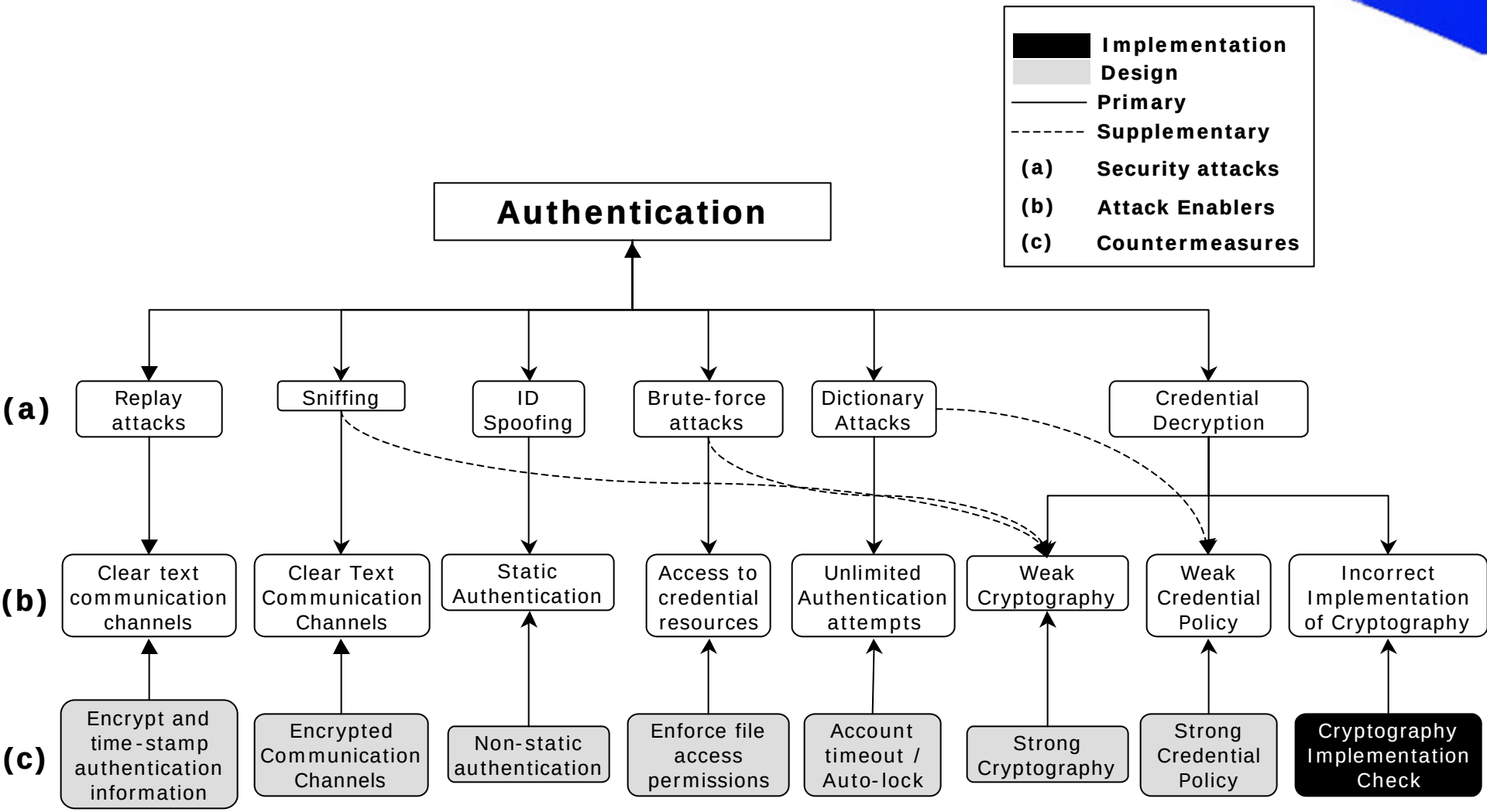
Methodology: Interacting Issues & Answers



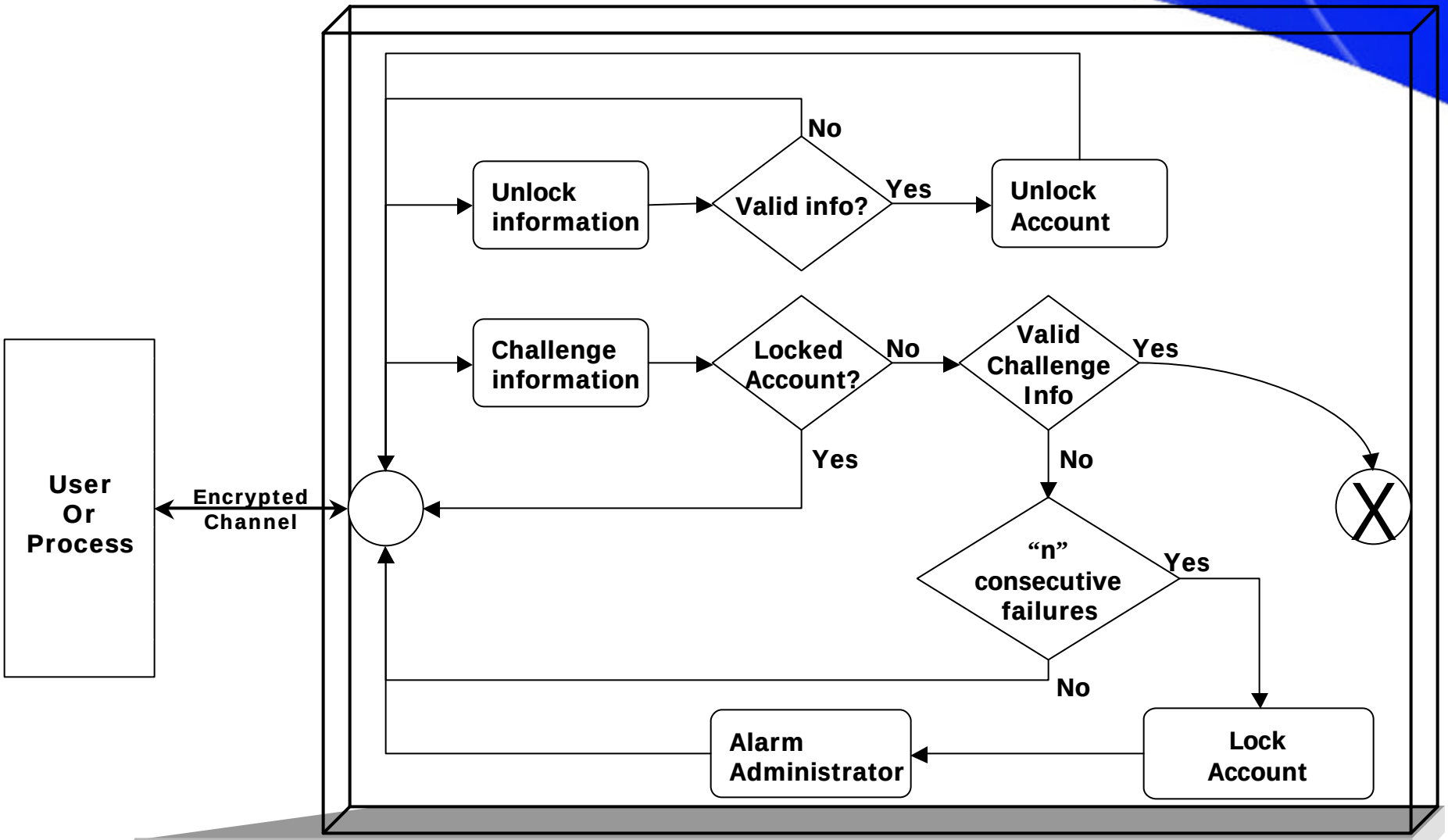
Our Methodology: Phase I

- **Phase 1: Select features and derive design models**
- Select security features.
- For each security feature, derive a countermeasures design model:
  - Identify and abstract all attacks related to the security feature.
  - For each security attack:
    - Derive all attack enablers.
    - For each attack enabler
      - Prescribe appropriate *security countermeasures*.
      - For each countermeasure:
        - Analyze countermeasure for residual vulnerabilities.
        - Add corrective measures to overcome vulnerabilities.
  - Derive the complete security-oriented countermeasures design model.
    - Group all prescribed countermeasures
    - Separate countermeasures into action countermeasures and underlying countermeasures.
    - Divide countermeasures design model into an action-box and underlying planes.
      - Put action countermeasures inside box [group into flowcharts]
      - Add Underlying Countermeasures
  - Verify attack coverage via traceability matrix.

Phase 1 Case Study: Authentication



Authentication Countermeasures Model



Authentication Design Audit

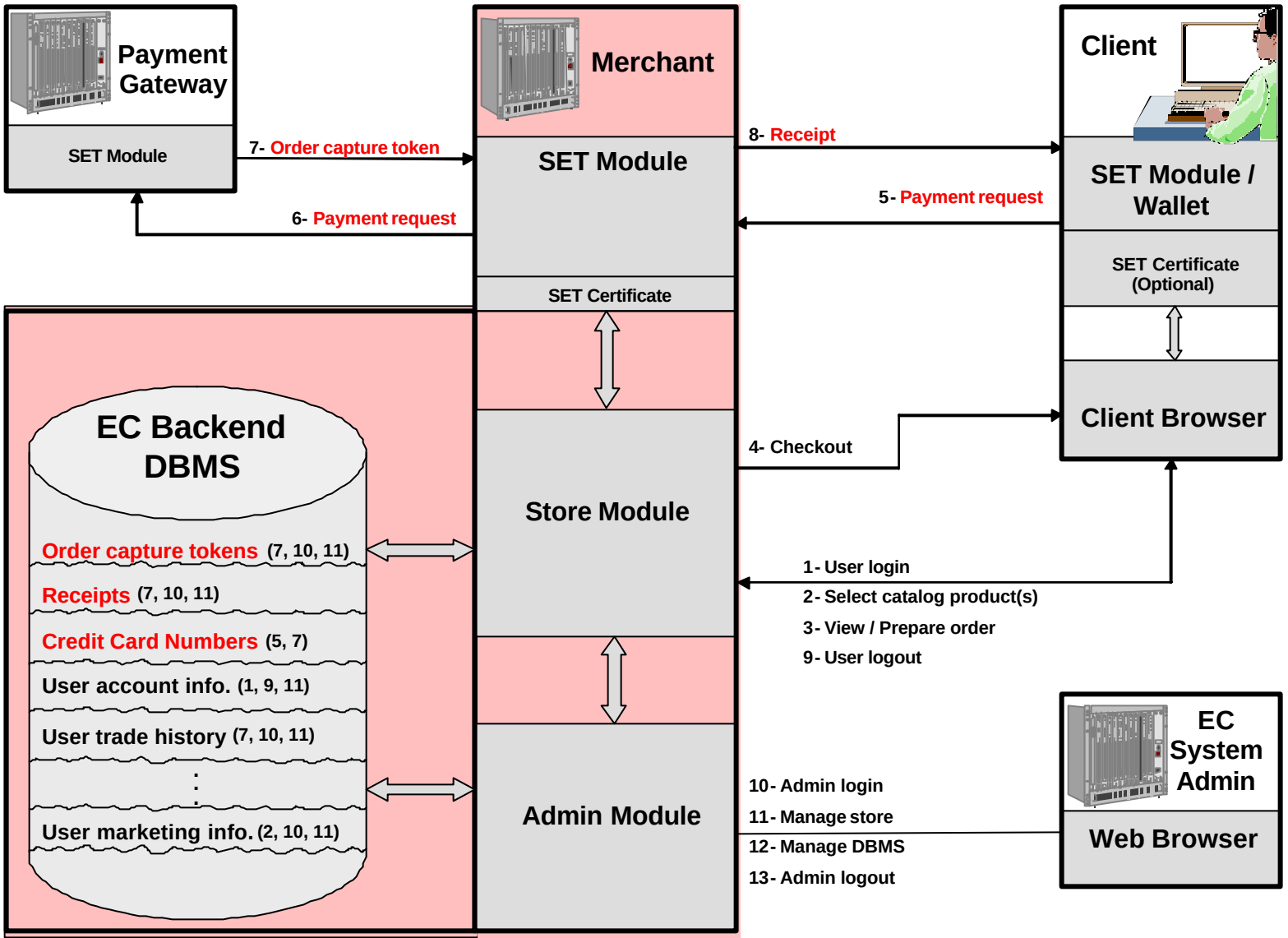
COUNTERMEASURES	AUTHENTICATION SECURITY ATTACK TYPE					
	Sniffing attacks	ID Spoofing attacks	Brute-force attacks	Dictionary attacks	Replay attacks	Credential Decryption
Encrypted comm. channels	D				D	
Non-static authentication		D				
Enforce file-access permissions			D			
Account timeout / Auto-lock				D		
Strong crypt.						D
Strong credential policy						D
Cryptography impl. check						I

D = design time, I = after system implementation

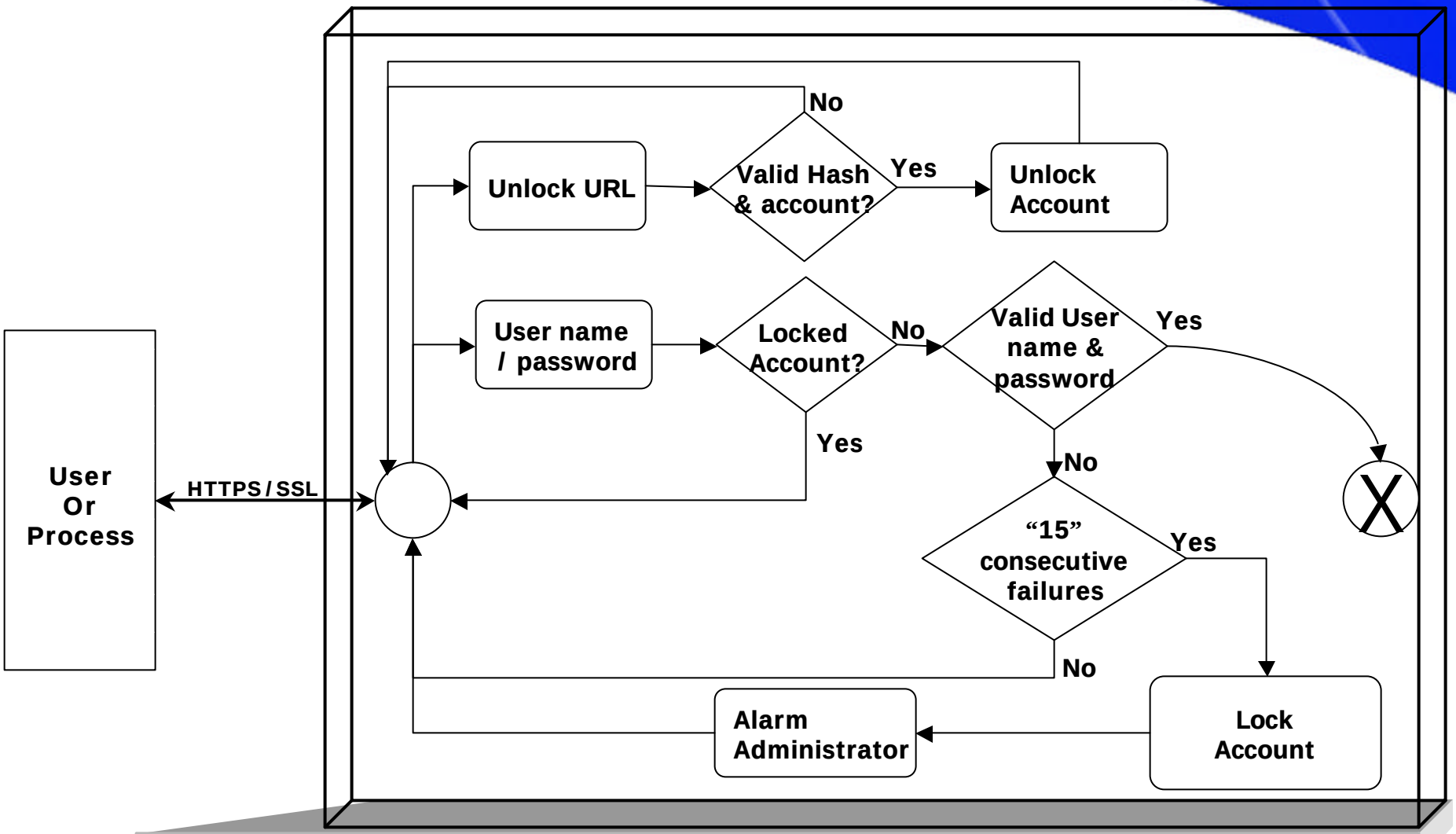
Our Methodology: Phase 2

- **Phase 2: Instantiate and Integrate the derived models into an e-commerce system design**
  - Instantiate the Countermeasures Design Models for Each Security Feature
  - Integrate the Instantiated Models into an Existing Design

Phase 2 Case Study: A SET-Integrated EC System



Phase 2: Instantiated Authentication Model



Benefits of:

Methodology

- Systematic and modular in nature
- Satisfies standard user requirements at system design time
- Specializes the security features of the NIST security services model without breaking any interdependent relationships among these features
- Effectively addresses and blocks all known malicious user requirements
- Provides implementation and testing guidelines
- Better chances of achieving overall security in e-commerce systems

Countermeasures Models

- Extensible
- Easily instantiated and integrated into HLD
- Can be used with security-aware technologies such as SET
- Phase 2 can be repeated for a variety of EC designs without repeating Phase 1

Limitations of:

Methodology

- Does not provide countermeasures against future attacks
- Not systematic when analyzing residual vulnerabilities of prescribed countermeasures
- Prescribed countermeasures might impact performance

Countermeasures Models

- The instantiation process is not systematic in nature
- It does not explicitly state and model the relationship between the various countermeasures introduced

Contributions of our Approach (so far)

1. A design audit matrix mapping all known security attacks on four security features
2. Four new security models that extend the NIST security services model for e-commerce systems
3. A cost-effective, systematic methodology for deriving countermeasures design models for the other security features of e-commerce systems (4 done, others for future work)
4. An overview of all known security attacks related to the four security features

Future Work

- Apply our methodology and approach to the remaining features of the NIST security services model.
- Enhance the methodology to map other security-related features, such as impact on performance, into the design process.
- Formally describe the methodology and provide tools.
- Develop a Security Design Handbook and teach this approach to E-Business-Based Information Systems Analysts, Designers, and Developers!

Thank You!