

Science of Cyber Security: Modeling, Composition, and Measurement

Andre Scedrov (co-PI), John Mitchell (PI)
U Penn Stanford

Stanford, Berkeley, Carnegie-Mellon, Cornell, U Penn

Project Goals

- ▶ **Scientific objective**

- ▶ Advance the science base for trustworthiness by developing concepts, relationships, and laws with predictive value.

- ▶ **Technical approach**

- ▶ *Security modeling*: characterize system, threats, and desired properties. Leverage game-theoretic concepts to model incentives for the defender and attacker.
 - ▶ *Composition*: develop principles for explaining when security schemes compose, and how to achieve compositionality.
 - ▶ *Security Measurement*: goals include determining relative strengths of defense mechanisms, evaluating design improvements, and calculating whether additional mechanism is warranted based on attacker and defender incentives



Team

- ▶ Anupam Datta (CMU)
- ▶ Joe Halpern (Cornell)
- ▶ John Mitchell (Stanford, PI)
- ▶ Andrew Myers (Cornell)
- ▶ Andre Scedrov (U Penn)
- ▶ Fred Schneider (Cornell)
- ▶ David Wagner (UC Berkeley)
- ▶ Jeannette Wing (CMU)
- ▶ Ittai Abraham (Microsoft Research, unfunded collaborator)

Contact: mitchell@cs.stanford.edu



Laws: Analysis and Synthesis

- ▶ **Analysis:** Given an artifact, predict its properties...
 - ▶ Qualitative properties: What it does?
 - ▶ Quantitative properties: How well?
- ▶ **Synthesis:** Compose artifacts with given properties to obtain a new one with predictable properties.

Project Goals

- ▶ **Scientific objective**

- ▶ Advance the science base for trustworthiness by developing concepts, relationships, and laws with predictive value.

- ▶ **Technical approach**

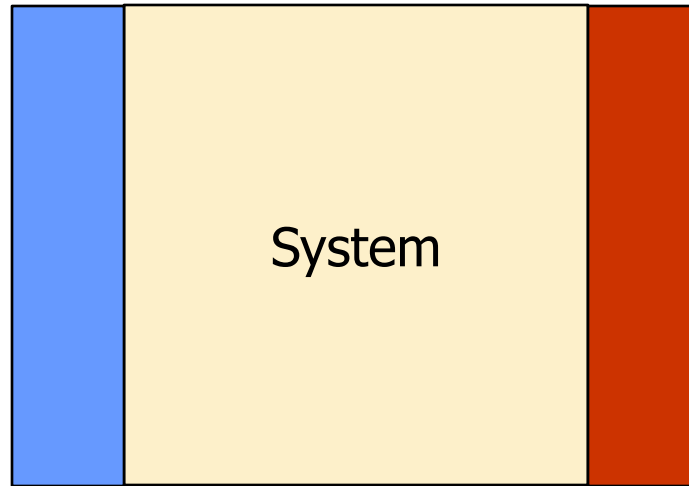
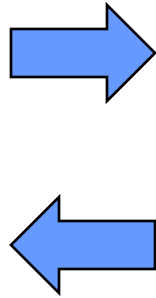
- ➡ *Security modeling*: characterize system, threats, and desired properties. Leverage game-theoretic concepts to model incentives for the defender and attacker.
 - ▶ *Composition*: develop principles for explaining when security schemes compose, and how to achieve compositionality.
 - ▶ *Security Measurement*: goals include determining relative strengths of defense mechanisms, evaluating design improvements, and calculating whether additional mechanism is warranted based on attacker and defender incentives



Security modeling and analysis: System, Properties, Threat model



User



Attacker

- ▶ Users interact with as system of interest
- ▶ System is intended to have some set of desirable properties
 - ▶ If users enter expected input, system will deliver desired output
- ▶ Attacker may have a different *interface* and *capabilities*. May:
 - ▶ Disrupt honest user's use of the system
 - ▶ Learns information intended for others only

Measurement

- ▶ Threat models can be compared
 - ▶ Every mechanism that is secure against a stronger threat model is secure against a weaker one
- ▶ Properties can be compared
 - ▶ Every mechanism that guarantees a stronger property also guarantees a weaker one
- ▶ Quantitative measures are possible
 - ▶ Concrete resource bounds on adversary
 - ▶ Incentives parameterized by quantitative payoffs
 - ▶ Size and strength of defenses
 - ▶ Number of repeated “plays of the game”



Modeling system users and adversaries

- ▶ Recent work by psychologists aims to find models that predicted human behavior well.
 - ▶ Example: Human subjects were given a choice between making a “risky” move and a safe move.
 - ▶ Repeated in different settings, with different parameters
 - ▶ Challenged other researchers to find a model whose predictions best match the data
 - ▶ Promising approach: inertia, sampling, and weighting
 - ▶ Exploration – Try a different action.
 - ▶ Inertia – keep doing what you did before, unless you are “surprised”.
 - ▶ Exploitation - play the action with the best estimated subjective value, given previous outcomes
-

Science of Security project

- ▶ The *I-SAW* model seems intuitive, but it is unclear if it is rational.
- ▶ We show that it can be viewed as the outcome of play of a resource-bounded agent modeled as a finite automaton. [Seeman, Halpern, Pass]
- ▶ We describe a model and a simple strategy for a bounded agent, and show both theoretically and by simulation that it does very well in practice.
- ▶ Importance: precise automata-theoretic model that can represent an abstraction of user behavior, for studying security of a variety of systems.



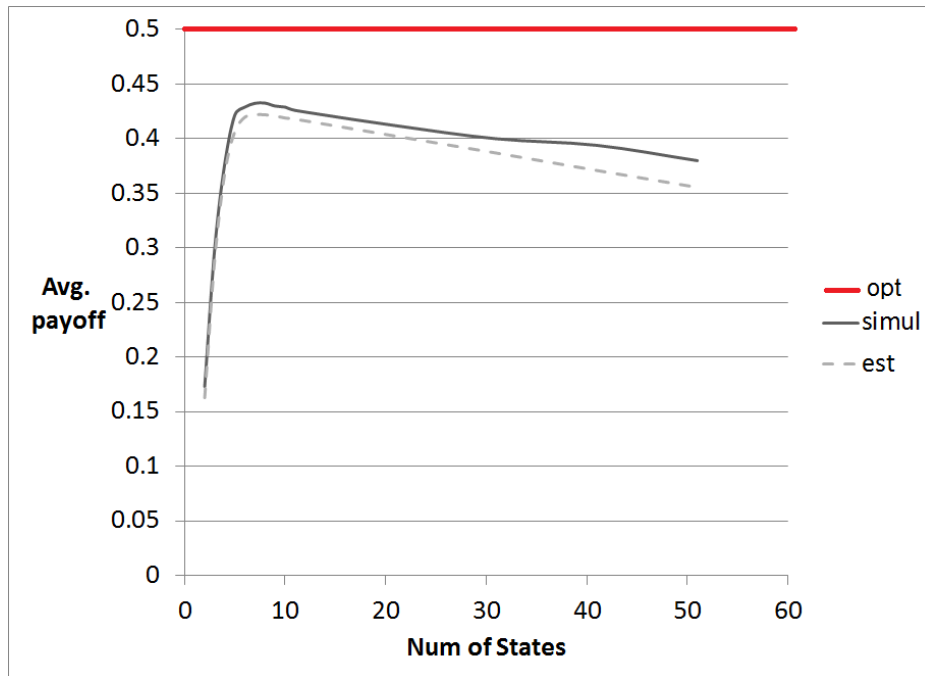
Simulation Result

Our theoretical analysis shows that the automaton performs well as the number of automaton states goes to infinity and changes of nature's state become rarer.

- ▶ But what happens if the automaton has only a few states, for a reasonable transition probability for nature?
 - ▶ Our objective was to show that this automaton performs well even with a small number of states and with a reasonable transition probability π .
- ▶ We simulated this model with signals prob. (.4,.3,.2,.1) ,payoffs of 1,-1 and $\pi=0.001$.



Simulation Result



- ▶ This graph shows that even with a few states the agent gets close to the optimal payoff of 0.5.
- ▶ With only 4 states he already gets more than 0.4.

It also shows the theoretical lower bound is a good estimator.



Measuring information leakage

- ▶ Traditional approaches to quantitative information flow measure information as uncertainty:
 - ▶ Leakage = uncertainty a priori – uncertainty a posteriori
- ▶ Different measures of uncertainty capture different notions of leakage:
 - ▶ **Shannon entropy**: how many bits of the secret are leaked?
 - ▶ **Min-entropy**: how likely it is that the secret will be guessed?
 - ▶ **Guessing entropy**: how long does it take for the secret to be guessed?
- ▶ Entropy measures are indifferent to the **meaning** of bits:
 - ▶ Ex: In a bank system, the bits representing the user's street address are less sensitive than the bits representing his account number and pin-code. Even if both fields may have the same number of bits, the **value** leaked is not the same.



Taking meaning and value into consideration

- ▶ Goal: to incorporate the notion of value into measures of information and leakage.
 - ▶ Identify meaningful operational scenarios where value is relevant;
 - ▶ Define mathematically measures that capture these scenarios;
 - ▶ Investigate the mathematical properties of these metrics (expected leakage, maximum leakage, optimal attack strategies).
- ▶ Previous work: *g-leakage* [Alvim et al.] can capture the notion of value of bits, but it doesn't address scenarios where certainty about guesses is required, and does not directly address the composition of attacks.



Algebraic foundations for measures of value

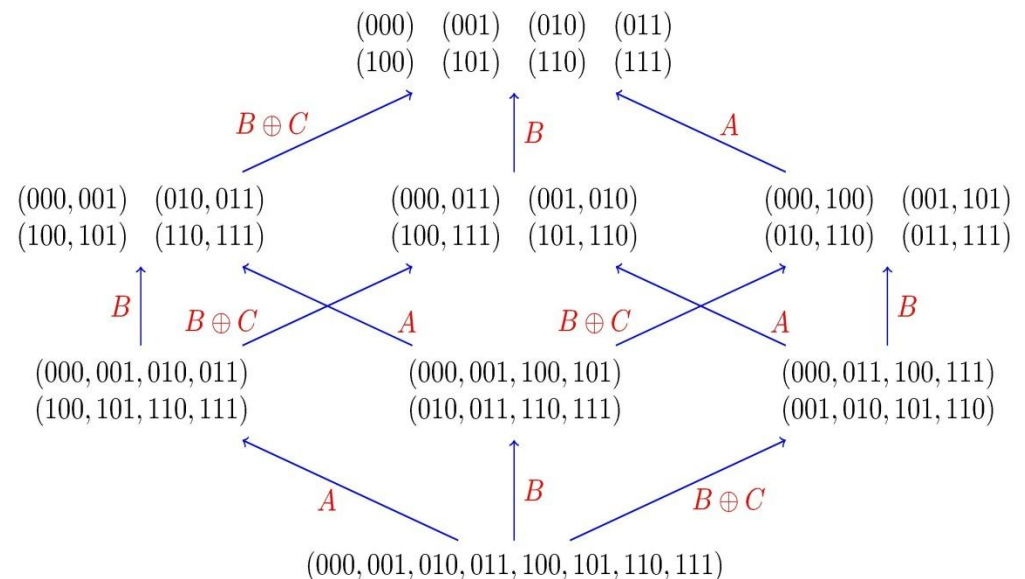
► The quantification of leakage goes through two steps:

► (1) Algebraic description of the system and the attacks

- Using the **lattice of information**: in a deterministic system, each attack induces a partition on the space of secrets, and the set of all partition forms a lattice.

► Example:

- secret of 3 bits (A,B,C)
- possible attacks/
observations (A, B, $B \oplus C$).



► (2) Numeric evaluation of each element of the lattice.

The mathematical properties of the measures of value

- ▶ The proposed measures of leakage must satisfy the lattice of information: for every measure of value v , and for any two states P_1 and P_2 of the program:

$$P_1 \sqsubseteq P_2 \Leftrightarrow v(P_1) \leq v(P_2)$$

- ▶ i.e. the higher an element is in the lattice, the more information it contains.
 - ▶ Proposed measures. Given an attack sequence:
 - ▶ **Valued deducibles**: what is the value of the bits that can be inferred for sure?
 - ▶ **Valued min-entropy**: what's the expected value that can be extracted in one attack?
 - ▶ **Valued guessability**:
 - ▶ How many attack steps are needed to achieve value v ?
-



Project Goals

- ▶ **Scientific objective**

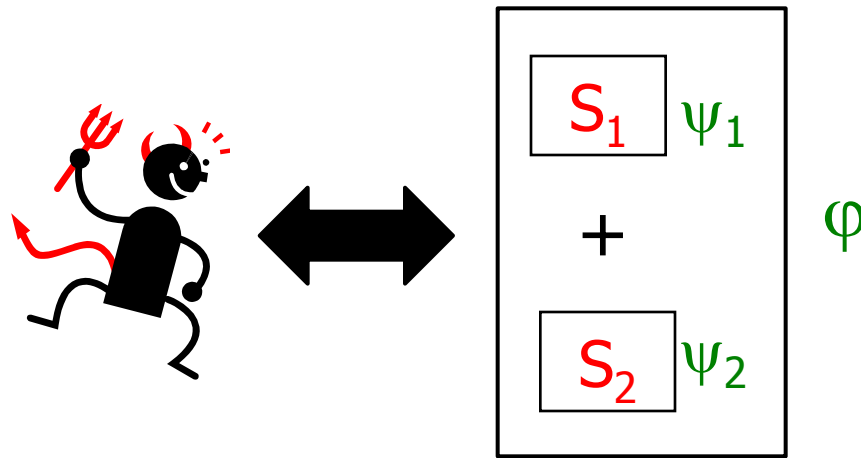
- ▶ Advance the science base for trustworthiness by developing concepts, relationships, and laws with predictive value.

- ▶ **Technical approach**

- ▶ *Security modeling*: characterize system, threats, and desired properties. Leverage game-theoretic concepts to model incentives for the defender and attacker.
 - ➡ *Composition*: develop principles for explaining when security schemes compose, and how to achieve compositionality.
 - ▶ *Security Measurement*: goals include determining relative strengths of defense mechanisms, evaluating design improvements, and calculating whether additional mechanism is warranted based on attacker and defender incentives



Goal: Compositional Security



- ▶ Do $S_1 + S_2$ satisfy a global security property ϕ based on local properties ψ_1 of S_1 and ψ_2 of S_2 that are checkable separately?
- ▶ Challenge: Adversary may supply data (handled in our prior work) and **code** (new challenge)

Compositional Security Results

- ▶ General programming language and type system to modularly reason about trace properties of higher-order functions [Datta, Garg, Jia 2012]
- ▶ Applications:
 - ▶ Proving code integrity and confidentiality properties of a hypervisor in the presence of adversaries that can modify code on disk
 - ▶ Proving existence and absence of inappropriate information flows in user-facing applications (e.g., when clicking on an email attachment sent by an adversary can leak information back to the adversary via browser call-back mechanisms)



Goal: Provenance in Composed Systems

► Provenance

- Definition: When something goes wrong in a composed software system, determine the relevant sequence of function calls that resulted in the final violating program step to be executed.
- Mechanism: Combining static program analysis and auditing. What information should be recorded in an audit log to enable provenance in higher-order interacting programs (e.g., browsers, OS, email clients)?



Toward General Compositional Security

- ▶ **Compositional security for higher-order systems**
 - ▶ Adversary can supply and modify code in addition to data
- ▶ **Secure abstraction-refinement principles**
 - ▶ Property-preservation from system design to software
- ▶ **Distributed systems (see next slides)**
 - ▶ Information flow
 - ▶ Timing channels
- ▶ **Applications**
 - ▶ Network protocols
 - ▶ Trusted computing systems + hypervisors
 - ▶ Web platform



Project Goals

- ▶ **Scientific objective**

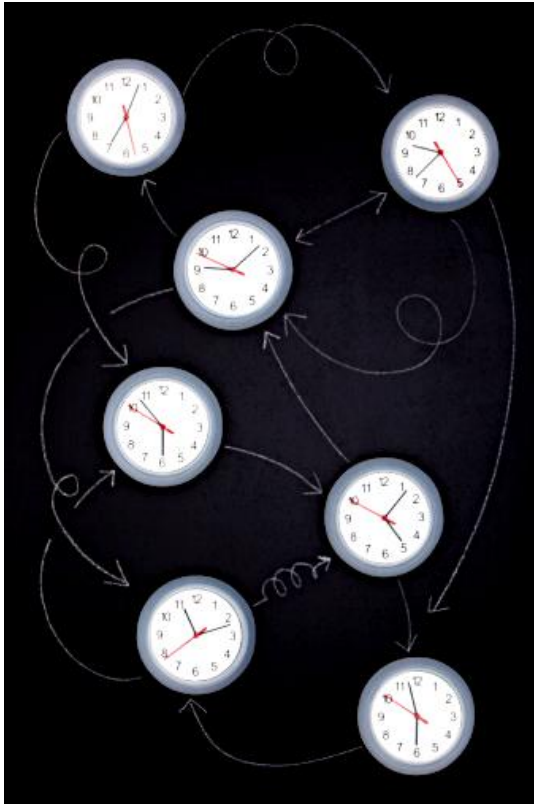
- ▶ Advance the science base for trustworthiness by developing concepts, relationships, and laws with predictive value.

- ▶ **Technical approach**

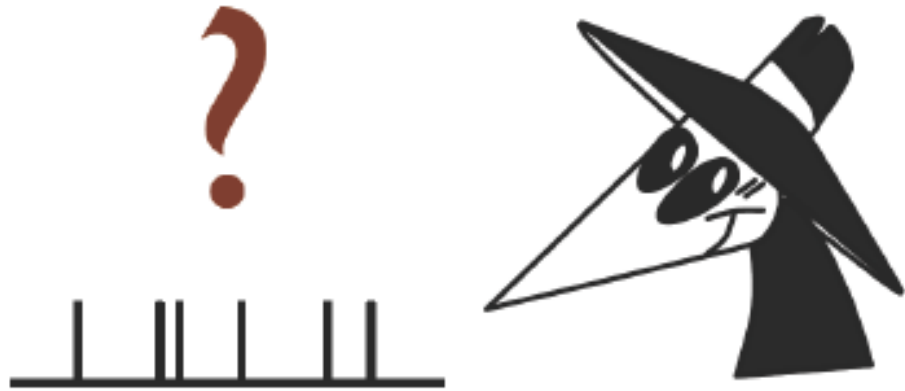
- ▶ *Security modeling*: characterize system, threats, and desired properties. Leverage game-theoretic concepts to model incentives for the defender and attacker.
 - ▶ *Composition*: develop principles for explaining when security schemes compose, and how to achieve compositionality.
 - ➡ *Security Measurement*: goals include determining relative strengths of defense mechanisms, evaluating design improvements, and calculating whether additional mechanism is warranted based on attacker and defender incentives



Timing channels



The adversary can learn (a lot) from timing measurements.



- ▶ Known to exist
- ▶ Hard to detect
- ▶ Hard to prevent except in special cases

A few timing attacks

- ▶ **Network timing attacks**

- ▶ RSA keys leaked by decryption time, measured across network [Brumley&Boneh'05]
- ▶ Load time of web page reveals login status, size and contents of shopping cart [Bortz&Boneh'07]

- ▶ **Cache timing attacks**

- ▶ AES keys leaked by timing memory accesses [Osvik et al'06] from ~300 (!) encryptions

- ▶ **Covert timing channels**

- ▶ Transmit confidential data by controlling response time, e.g., combined with SQL injection [Meer&Slaviero'07]

- ▶ **Timing channels : a serious threat**

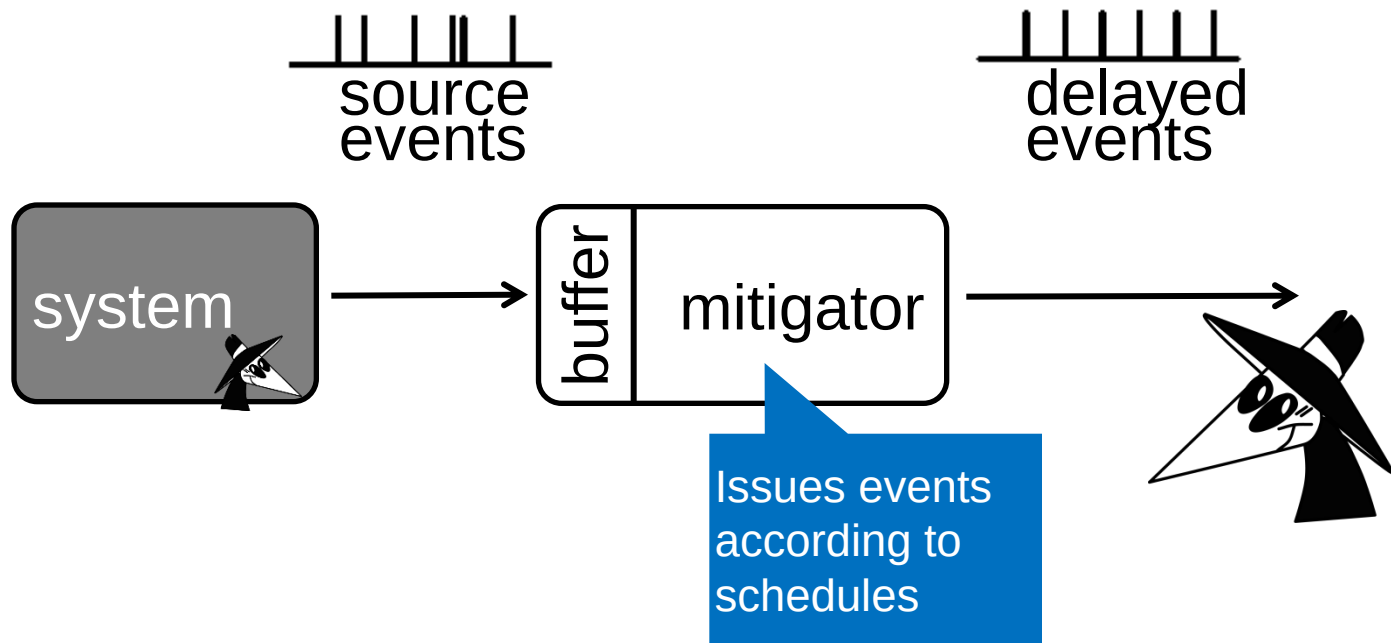


Timing channel mitigation

- ▶ Some standard ideas:
 - ▶ Add random delays $\Rightarrow$ lower bandwidth, linear leakage
 - ▶ Delay to worst-case time $\Rightarrow$ poor performance
 - ▶ Input blinding $\Rightarrow$ applicable only to cryptography
- ▶ New idea: predictive mitigation
 - ▶ Applies to general computation
 - ▶ Leakage asymptotically sublinear over time
 - ▶ Effective in practice
 - ▶ Applicable at system and language level



Black-box predictive mitigation



Project Goals

- ▶ **Scientific objective**

- ▶ Advance the science base for trustworthiness by developing concepts, relationships, and laws with predictive value.

- ▶ **Technical approach**

- ▶ *Security modeling*: characterize system, threats, and desired properties. Leverage game-theoretic concepts to model incentives for the defender and attacker.
 - ▶ *Composition*: develop principles for explaining when security schemes compose, and how to achieve compositionality.
 - ▶ *Security Measurement*: goals include determining relative strengths of defense mechanisms, evaluating design improvements, and calculating whether additional mechanism is warranted based on attacker and defender incentives



