FlipThem: Modeling Targeted Attacks with FlipIt for Multiple Resources

Aron Laszka[1], Gabor Horvath[2], Mark Felegyhazi[2], and Levente Buttyan[2]

[1]: Vanderbilt University, Institute for Software Integrated Systems
[2]: Budapest University of Technology and Economics, Department of Networked Systems and Services

Stealthy Attacks

- In many scenarios, attackers want to keep successful security compromises covert
- Examples

Cyber-espionage

- targets must **not know** that they are being spied on

Botnets

- users should **not be aware** that their computers are infected

Mitigating Cover Compromises

- Mitigation
  - possible losses can be minimized by resetting the computing resource into a known secure state
  - examples: changing a password or a private key, reinstalling a machine

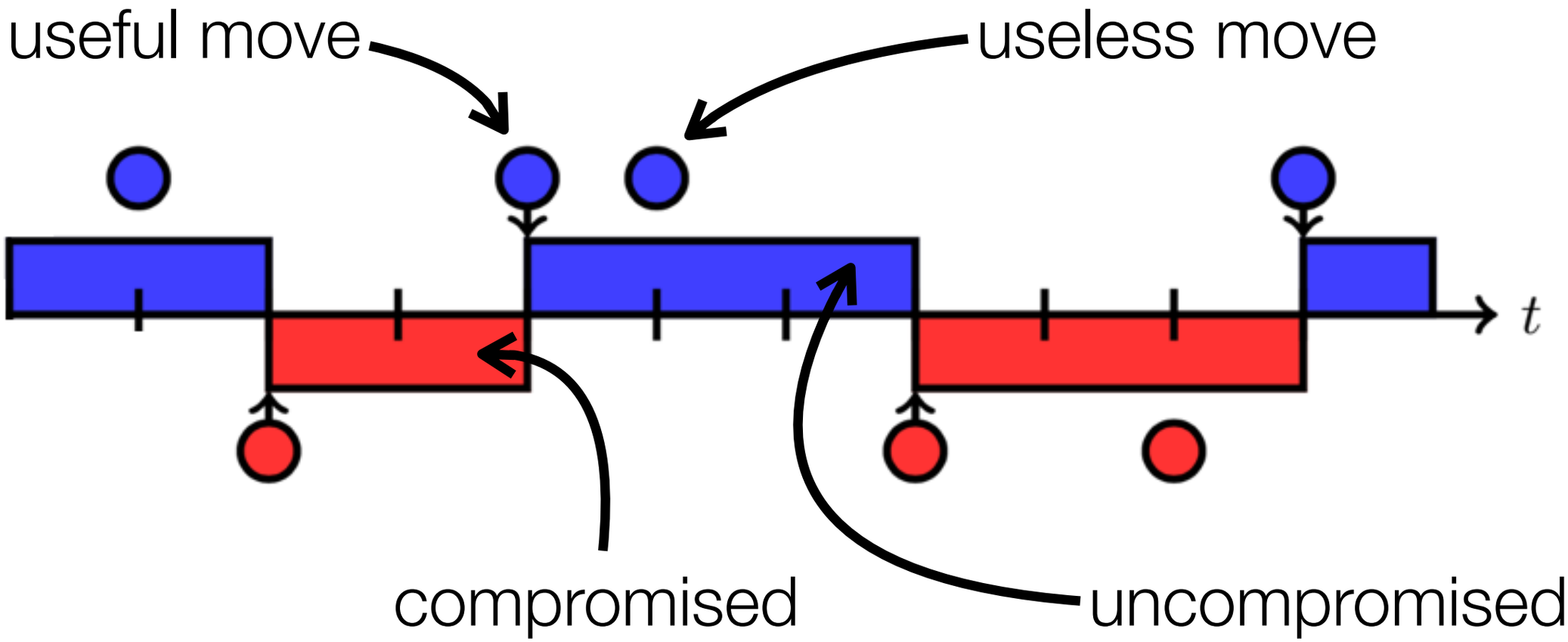
  “When should these moves be made?”

  - What is the optimal frequency?
  - What is the optimal scheduling?
- In practice: usually periodic key and password renewal strategies

The `FlipIt` Game

- Introduced by researchers at RSA for modeling stealthy attacks against computing resources
- Resource: user account, private key, machine, etc.
- Players
  - **defender**: the rightful owner of the resource
  - **attacker**: an adversary who is trying to take over the resource
- Strategy
  - schedule for a series of costly moves (e.g., periodic)
  - each move takes control of the resource (if it is not already controlled)
- Payoff: amount of time the resource is controlled by the player - cost of moves

The `FlipIt` Game - Graphical Illustration



Defender's payoff: 6 - 4 = 2

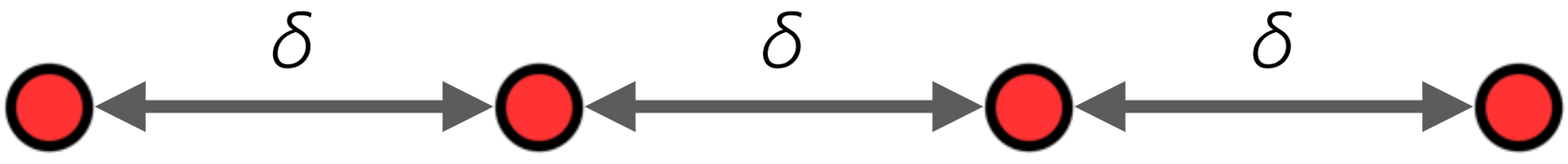
Attacker's payoff: 5 - 3 = 2

time controlled

moves

The `FlipIt` Game - Lessons Learned

- If there is no feedback, **periodic** strategies are dominant



- If the attacker learns the defender's previous moves when making a move,
  - then the defender is better off with a more random strategy, such as a renewal process with **exponential** interval distribution

  - for the attacker, **periodic** is still a good choice

Multiple Resources

- `FlipIt` tells us how to defend a single resource

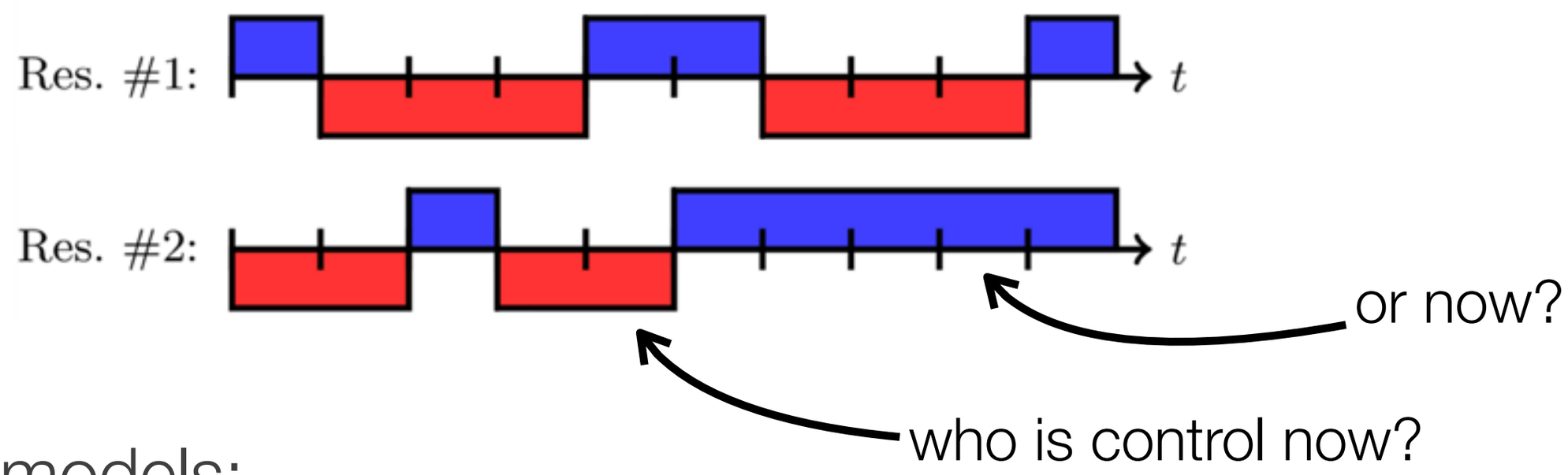
What if the security of a system depends on multiple resources?

- We could use a separate game for each resource

- But to exploit the dependencies between these resources, we need to model them together

Defining the Multiple-Resource Game

- Defining the players, the moves, etc. is straightforward
- Defining the payoffs is not straightforward



- Control models:

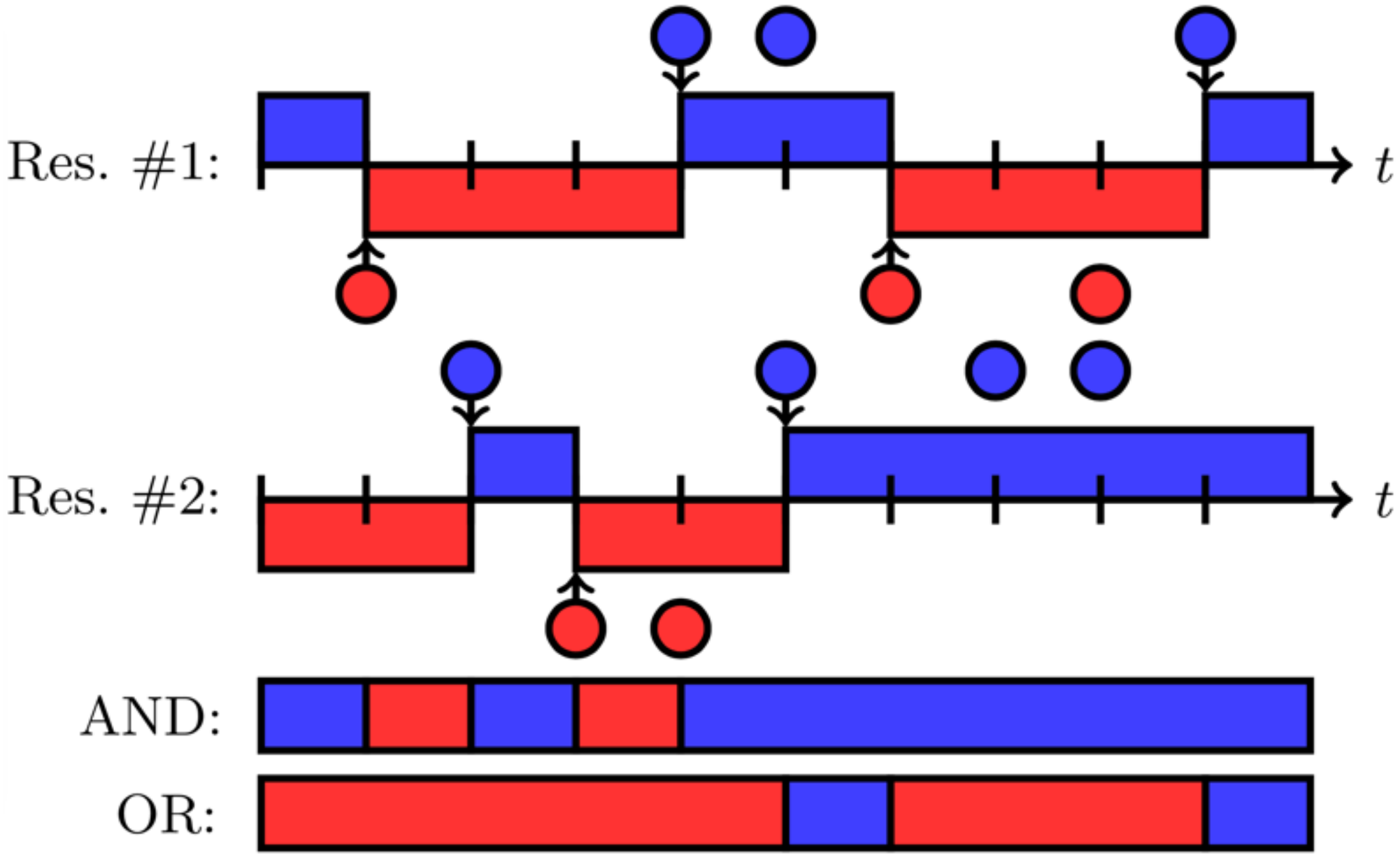
AND

attacker controls the system only if it controls *all* resources

OR

attacker controls the system if it controls *at least one* resource

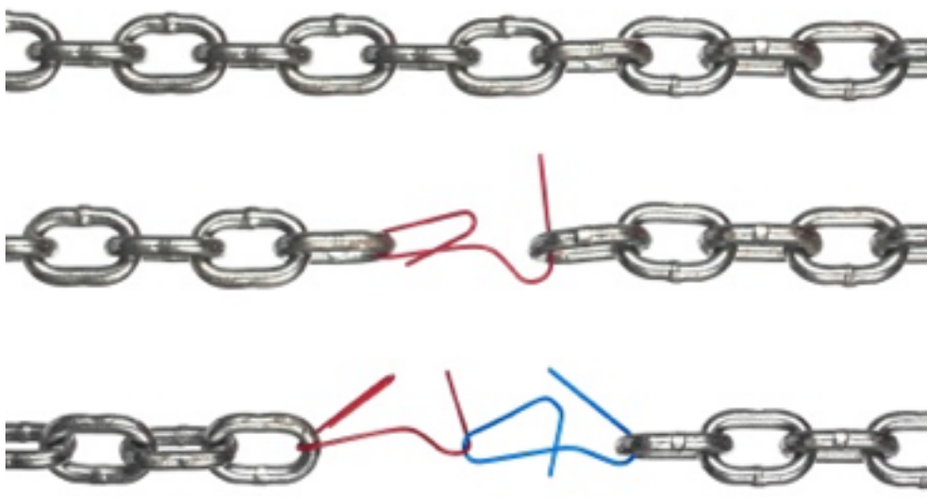
Illustration of Control Models



Control Models - Further Discussion

AND

- similar to the **total effort** model in security economics
- example: there are multiple private keys (stored separately), and the attacker needs to forge signatures for all of them
- defender is at advantage



OR

- similar to the **weakest link** model in security economics
- example: there are multiple administrator accounts on a machine, and the attacker needs to compromise only one
- attacker is at advantage

Combining Single-Resource Strategies

- Idea: build multiple-resource strategies from single-resource strategies that perform well in the `FlipIt` game
- Combinations:

Independent

- flip each resource independently of the others (i.e., use N independent single-resource strategies)

Synchronized

- always flip all resources together (i.e., use only one single-resource strategy for all the resources)

"Which one is better?"

- For which player?
- In which control model?

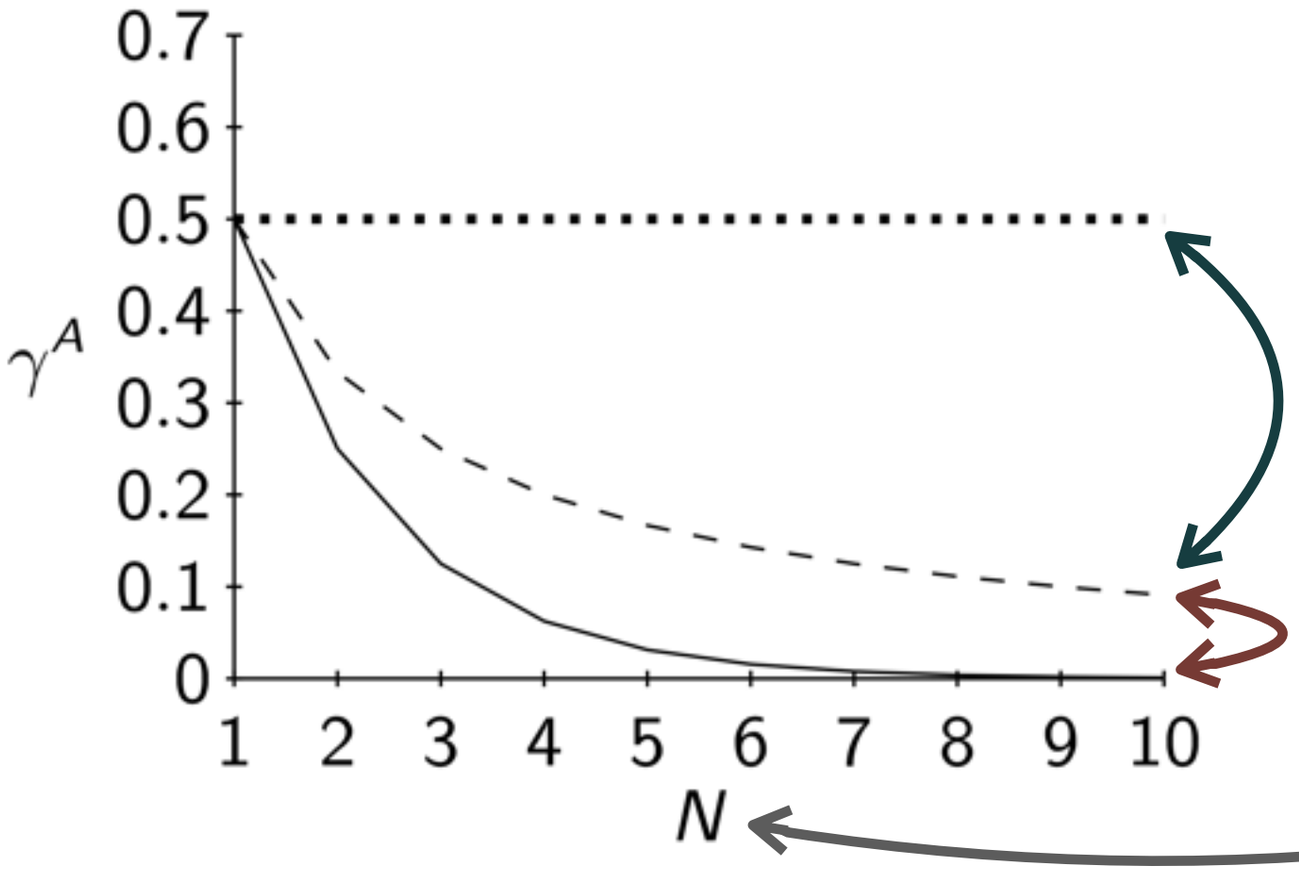
Attacker's Gain in the AND Model - Formulae #1

Defender's combination	Attacker's combination	Attacker's gain γ^A
independent	independent	$\prod_{r=1}^{N} \int_0^\infty f_{Z_r^D}(z_r) F_{Z_r^A}(z_r) dz_r$
	synchronized	$\int_0^\infty \prod_{r=1}^{N} \left(1 - F_{Z_r^D}(z)\right) f_{Z^A}(z) dz$
synchronized	synchronized	$\int_0^\infty f_{Z^D}(z) F_{Z^A}(z) dz$
	independent	$\int_0^\infty \prod_{r=1}^{N} F_{Z_r^A}(z) f_{Z^D}(z) dz$

Attacker's Gain in the AND Model - Formulae #2

Defender			Attacker			Attacker's gain
	single-res.	comb.		single-res.	comb.	γ^A
			strategy			
$\mathcal{E}$	ind.		$\mathcal{E}$	ind.		$\prod_{r=1}^{N} \frac{\alpha_r^A}{\alpha_r^A + \alpha_r^D}$
					syn.	$\frac{\alpha^A}{\alpha^A + \sum_{r=1}^{N} \alpha_r^D}$
	syn.					$\frac{\alpha^A}{\alpha^A + \alpha^D}$
	ind.		$\mathcal{P}$	ind.		$\prod_{r=1}^{N} \frac{\alpha_r^A}{\alpha_r^D} \left(1 - e^{-\frac{\alpha_r^D}{\alpha_r^A}}\right)$
					syn.	$\frac{\alpha^A}{\sum_{r=1}^{N} \alpha_r^D} \left(1 - e^{-\frac{\sum_{r=1}^{N} \alpha_r^D}{\alpha^A}}\right)$
	syn.					$\frac{\alpha^A}{\alpha^D} \left(1 - e^{-\frac{\alpha^D}{\alpha^A}}\right)$

Attacker's Gain in the AND Model - Numerical #1



defender should use independent

attacker should use synchronized

the more resources that have to be compromised, the safer the systems is

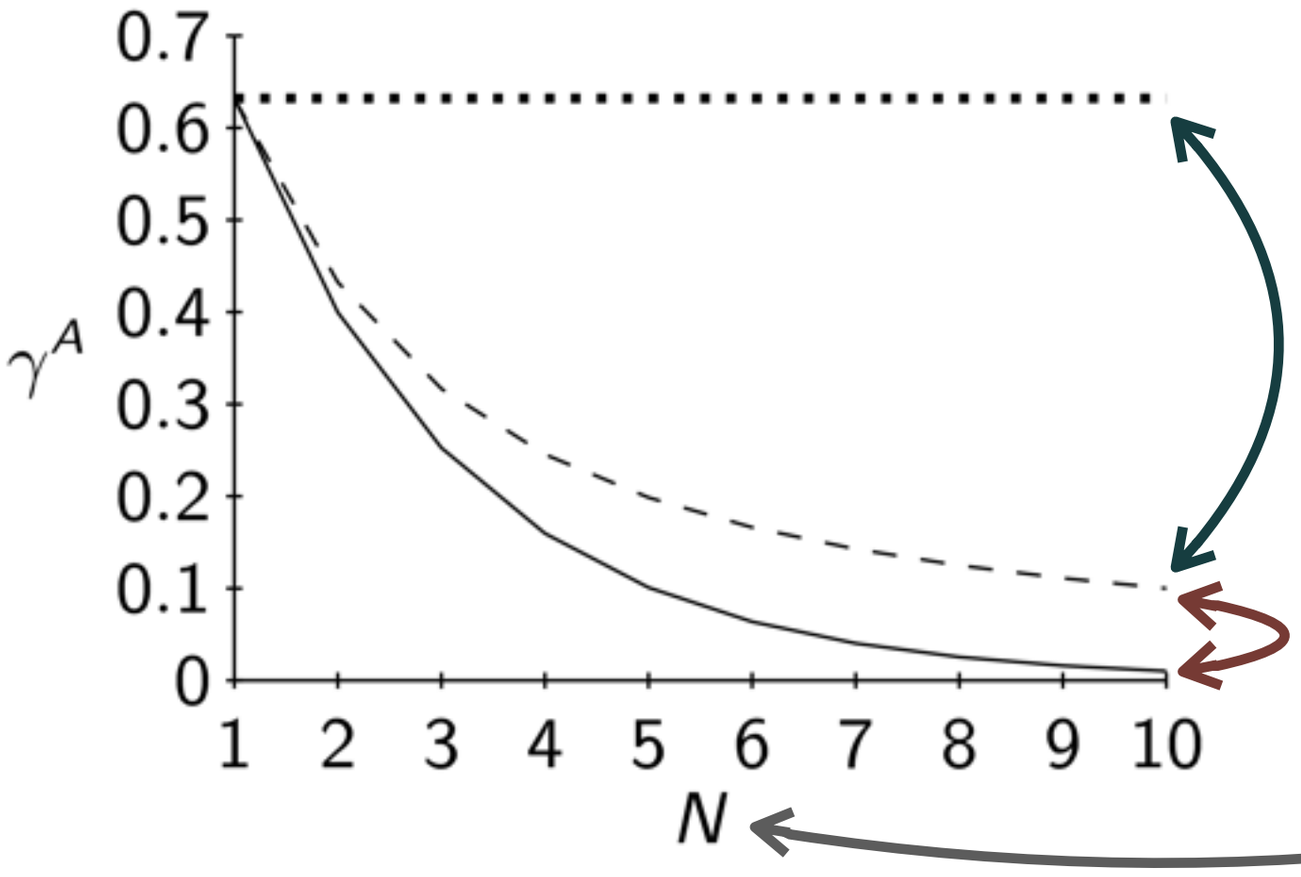
—— both players use independent strategies

- - attacker uses synchronized, while defender uses independent

···· both players use synchronized

(both players build on exponential single-resource strategies)

Attacker's Gain in the AND Model - Numerical #2



defender should use independent

attacker should use synchronized

the more resources that have to be compromised, the safer the systems is

—— both players use independent strategies

- - attacker uses synchronized, while defender uses independent

···· both players use synchronized

(defender builds on exponential, attacker builds on periodic single-resource strategies)

Strategy Combinations - Lessons Learned

- In the **AND** model,
  - **defender** should use **independent** strategies
  - **attacker** should use **synchronized** strategies

Since the two control models are the same with the roles of the players reversed, we readily have that

- in the **OR** model,
  - **defender** should use **synchronized** strategies
  - **attacker** should use **independent** strategies

Modeling assumptions matter a lot!

Markov Strategy Class

- Definition:
  at each time instance, the defender may flip any subset of the resources, and the probability of flipping a given subset depends on the times elapsed since flipping each resource
- "Multi-dimensional renewal process"
- Generalizes the above single-resource combinations
  - independent: probability of flipping a given resource depends on the time elapsed since last flipping that resource, and the probability of flipping a subset is simply the product of its elements' probabilities
  - synchronized: either all resources are flipped or none are, and the probability depends on the time elapsed since the last flip

Markov Strategies - Linear Programming Solution

- We assume that intervals given by the strategy are
  - discrete (e.g., key or password renewal policy is defined in days or weeks)
  - finite (i.e., every key or password is changed eventually)

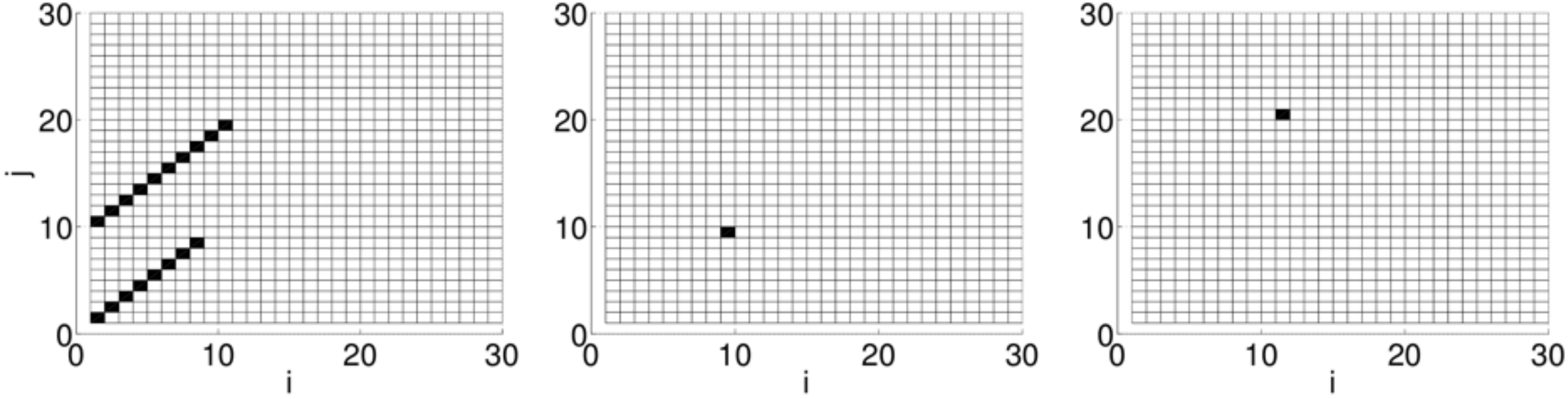
→ Markov strategy is defined by a finite set of probabilities

  - one for each subset of resources and each combination of times elapsed: (for example, with two resources, $p^S_{i,j}$ is the probability of flipping subset S given that the first resource was flipped i steps ago and the second resource was flipped j steps ago)

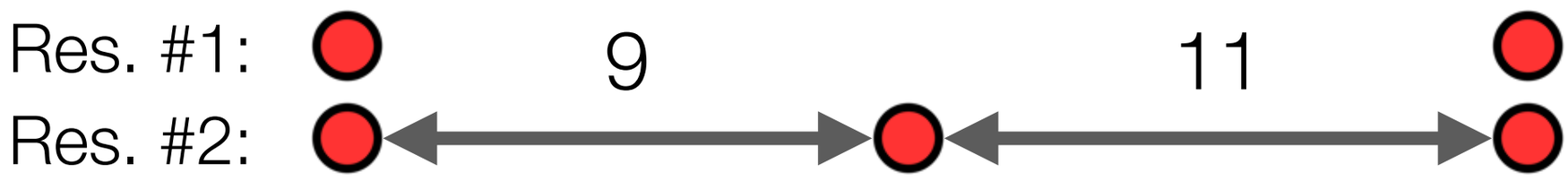
- For a given strategy, we can find the optimal best-response Markov strategy using linear programming
  - running time is exponential in the number of resources ☹
  - on a desktop PC, easy for a few resources and dozens time intervals ☺

Example: Markov Attack against a Given Defense

- Defender uses two independent exponential strategies with mean intervals 1 and 1/3
- Time steps are 0.03 long and the maximum number of time steps between two flips is 30

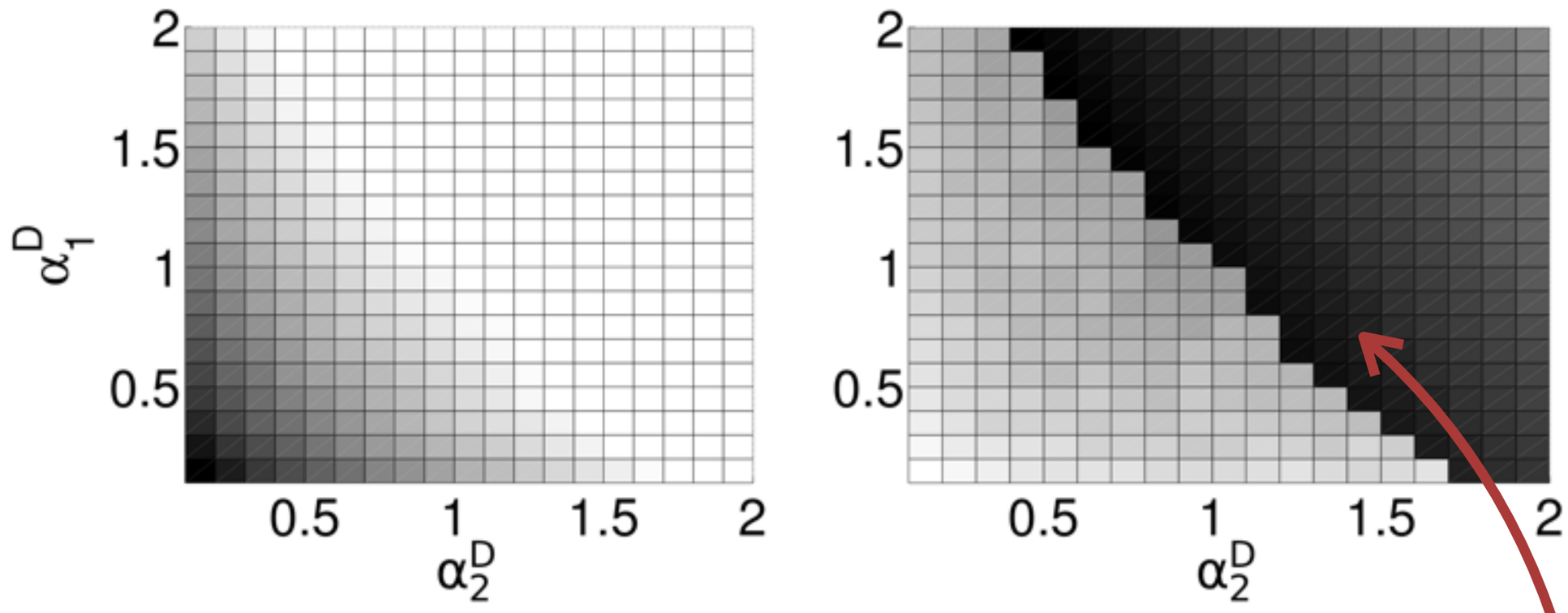


flip none | flip the second | flip both



Defense against a Markov Attacker (AND Model)

- Defender uses independent periodic strategies



Attacker's utility

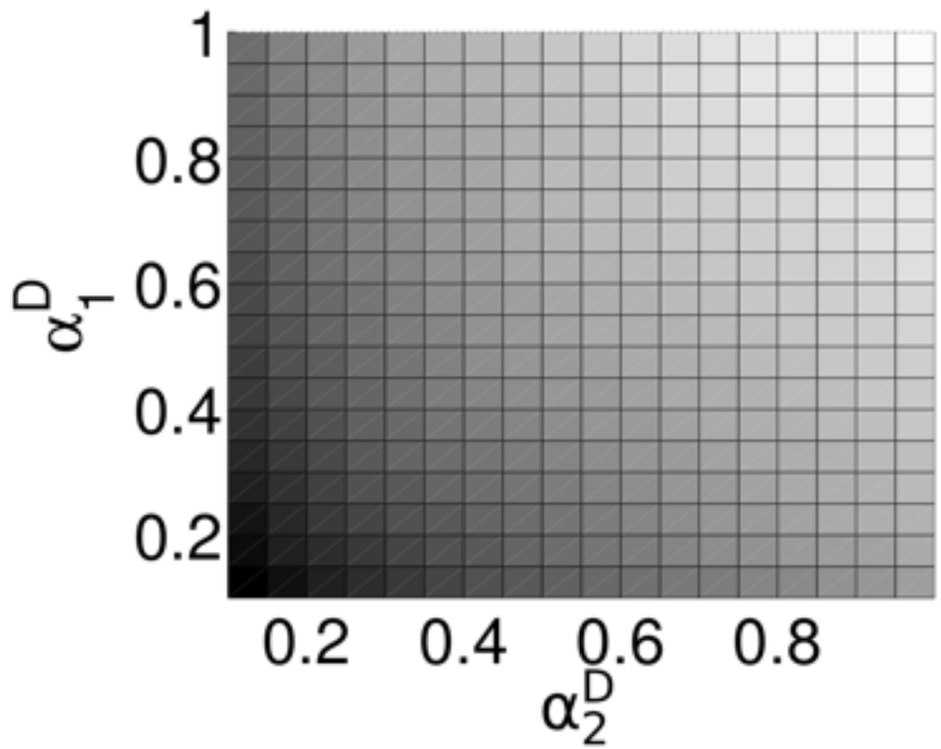
Defender's utility

attacker is deterred

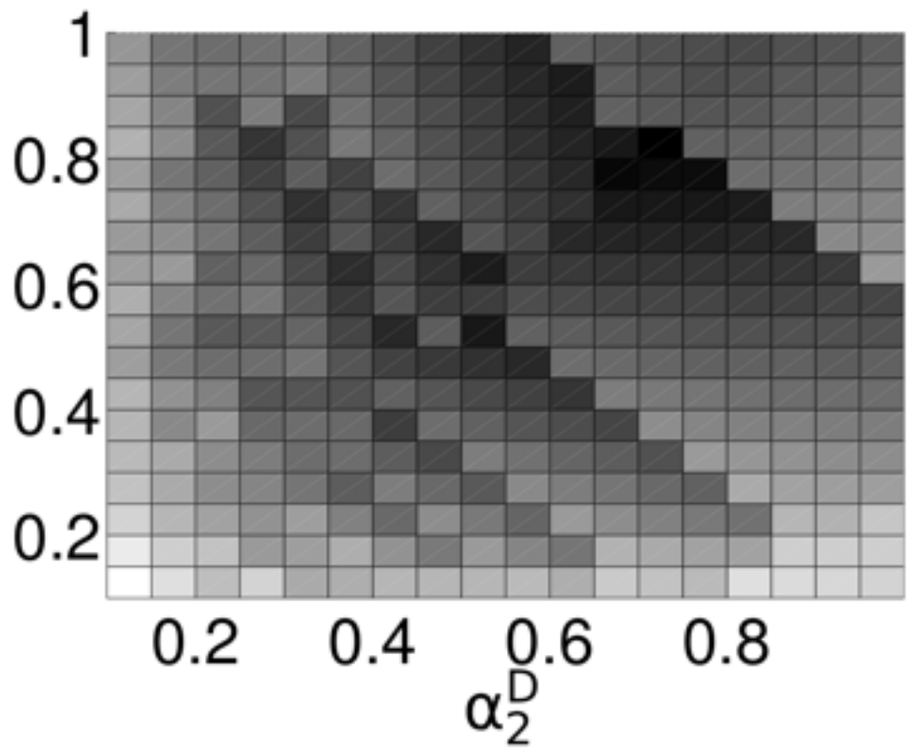
α^D_i: move rate for resource i
(darker shades represent higher utilities)

Defense against a Markov Attacker (AND Model)

- Defender uses independent exponential strategies



Attacker's utility



Defender's utility

α^D_i: move rate for resource i
(darker shades represent higher utilities)

Defense against a Markov Attacker - Lessons Learned

- Against a non-adaptive attacker, **independent periodic** strategies are good a choice in the AND model
  - however, an adaptive attacker could exploit this strategy
- Defender's utility is neither a continuous nor a monotonic function of the flipping rates, which makes optimization **challenging**
  - after the attacker has been deterred, increasing flipping rates only increases moving costs
  - with exponential strategies, the defender's utility has multiple local maxima

Thank you for your attention!

Questions?