Illegitimi non carborundum

Ronald L. Rivest

Viterbi Professor of EECS
MIT, Cambridge, MA

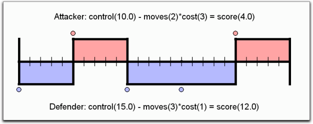
CRYPTO 2011
2011-08-15

Illegitimi non carborundum
(Don't let the bastards grind you down!)

Ronald L. Rivest

Viterbi Professor of EECS
MIT, Cambridge, MA

CRYPTO 2011
2011-08-15



Illegitimi non carborundum
(Don't let the bastards grind you down!)

Ronald L. Rivest

Viterbi Professor of EECS
MIT, Cambridge, MA

CRYPTO 2011
2011-08-15

Outline

Cryptography

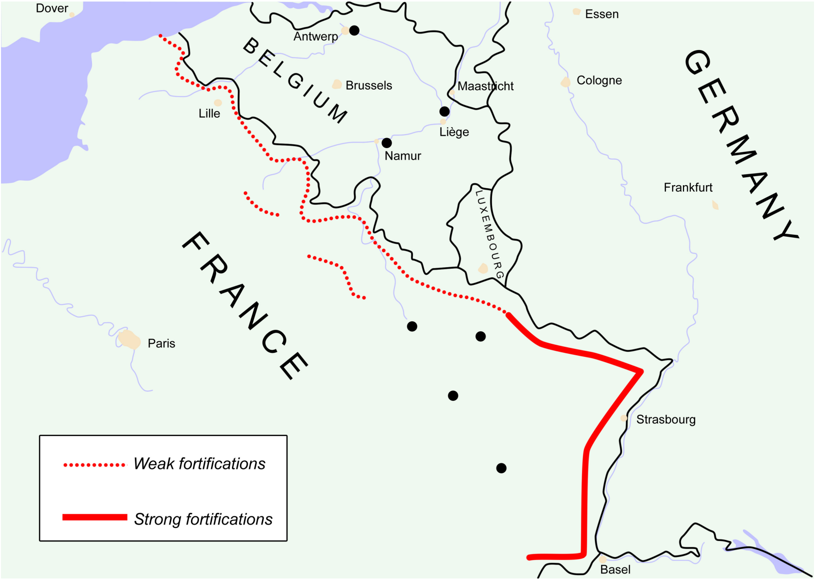
Cryptography is mostly about using *mathematics* and *secrets* to achieve confidentiality, integrity, or other security objectives.

Assumptions

We make *assumptions* as necessary, such as ability of parties to generate unpredictable keys and to keep them secret, or inability of adversary to perform certain computations.

Murphy's Law: "If anything can go wrong, it will!"

Assumptions may fail, badly. (Maginot Line)



Even worse...

In an adversarial situation, assumption may fail *repeatedly*...

(ref Advanced Persistent Threats)

Most crypto is like Maginot line...

We work hard to make up good keys and distribute them properly, then we sit back and wait for the attack.

There is a line we assume adversary can not cross (theft of keys).

Partial key theft

Much research allows adversary to steal *some portion* of key(s).

- secret-sharing [S79,...]
- proactive crypto [HJKY95,...]
- signer-base intrusion-resilience [IR04,...]
- leakage-resilient crypto [MR04,...]

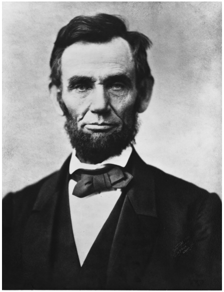
But adversary isn't allowed to steal *everything*, all at once. (Some exceptions, e.g. intrusion-resilient secure channels [IMR'05])

This just moves the line in the digital sand a bit...

Total key loss

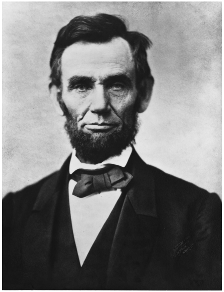
To be a good security professional, there shouldn't be limits on your paranoia!
(The adversary won't respect such limits...)
Are we being sufficiently paranoid??

Lincoln’s Riddle



Q: “If I call the dog’s tail a leg, how many legs does it have?”

Lincoln's Riddle



Q: "If I call the dog's tail a leg, how many legs does it have?"

A: "Four. It doesn't matter what you *call* the tail; it is still a tail."

Corollary to Lincoln's Riddle

Calling a bit-string a "secret key" doesn't actually make it secret...

Corollary to Lincoln's Riddle

Calling a bit-string a "secret key" doesn't actually make it secret...

Rather, it just identifies it as an interesting target for the adversary!

Our goal

To develop new models for scenarios involving total key loss.
Especially those scenarios where theft is *stealthy* or *covert*
(not immediately noticed by good guys).

The Game of "FLIPIT"
(aka "Stealthy Takeover")

joint work with
Ari Juels, Alina Oprea, Marten van Dijk
of RSA Labs

FLIPIT is a two-player game

- Defender = Player 0 = Blue
- Attacker = Player 1 = Red

FLIPIT is a two-player game

- Defender = Player 0 = Blue
- Attacker = Player 1 = Red

FLIPIT is rather symmetric, and we say "player *i*" to refer to an arbitrary player.

There is a contested critical secret or resource

There is a contested critical secret or resource

Examples:

- A password

There is a contested critical secret or resource

Examples:

- A password
- A digital signature key

There is a contested critical secret or resource

Examples:

- A password
- A digital signature key
- A computer system

There is a contested critical secret or resource

Examples:

- A password
- A digital signature key
- A computer system
- A mountain pass

State of secret or resource is binary

Good | Bad

State of secret or resource is binary

Good	Bad
Secret	Guessed or Stolen

State of secret or resource is binary

Good	Bad
Secret	Guessed or Stolen
Clean	Compromised

State of secret or resource is binary

Good	|	Bad
Secret	|	Guessed or Stolen
Clean	|	Compromised
Controlled by Defender	|	Controlled by Attacker

State of secret or resource is binary

Good	|	Bad
Secret	|	Guessed or Stolen
Clean	|	Compromised
Controlled by Defender	|	Controlled by Attacker
Blue	|	Red

A player can “move” (take control) at any time

- Defender move puts resource into Good state

A player can “move” (take control) at any time

- Defender move puts resource into Good state
  $=$ Initialize Reset Recover Dis-infect

A player can “move” (take control) at any time

- Defender move puts resource into Good state
  $=$ Initialize Reset Recover Dis-infect

- Attacker move puts resource into Bad state

A player can “move” (take control) at any time

- Defender move puts resource into Good state
  $=$ Initialize Reset Recover Dis-infect

- Attacker move puts resource into Bad state
  $=$ Compromise Corrupt Steal Infect

A player can “move” (take control) at any time

- Defender move puts resource into Good state
  $=$ Initialize Reset Recover Dis-infect

- Attacker move puts resource into Bad state
  $=$ Compromise Corrupt Steal Infect

Time is *continuous*, not discrete.

A player can "move" (take control) at any time

- Defender move puts resource into Good state
  $=$ Initialize Reset Recover Dis-infect

- Attacker move puts resource into Bad state
  $=$ Compromise Corrupt Steal Infect

Time is *continuous*, not discrete.
Players move at same time with probability 0.

Examples of moves:

- Create new password or signing key.
- Steal password or signing key.

Examples of moves:

- Create new password or signing key.
- Steal password or signing key.

- Re-install system software.
- Use zero-day attack to install rootkit.

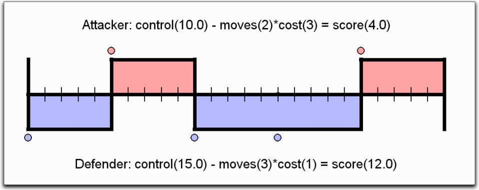
Examples of moves:

- Create new password or signing key.
- Steal password or signing key.

- Re-install system software.
- Use zero-day attack to install rootkit.

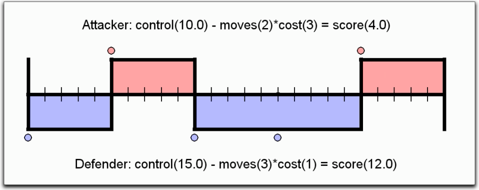
- Send soldiers to mountain pass.
- Send soldiers to mountain pass.

Continual back-and-forth warfare...



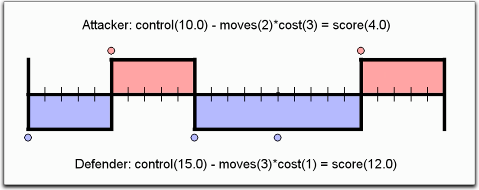
- Note that Attacker can take over at any time.

Continual back-and-forth warfare...



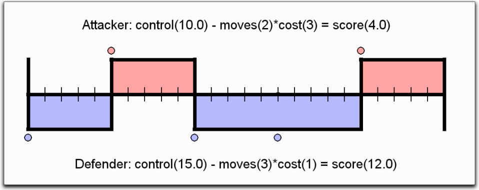
- Note that Attacker can take over at any time.
- There is no “perfect defense”.

Continual back-and-forth warfare...



- Note that Attacker can take over at any time.
- There is no “perfect defense”.
- Only option for Defender is to re-take control later by moving again.

Continual back-and-forth warfare...



- Note that Attacker can take over at any time.
- There is no “perfect defense”.
- Only option for Defender is to re-take control later by moving again.
- The game may go on forever...

Moves are “stealthy”

- In practice, compromise is often undetected...

Moves are “stealthy”

- In practice, compromise is often undetected...
- In FLIPIT,
  players do *not* immediately know when the other player makes a move!
  (Very unusual in game theory literature!)

Moves are “stealthy”

- In practice, compromise is often undetected...
- In FlipIt,
  players do *not* immediately know when the other player makes a move!
  (Very unusual in game theory literature!)
- Player’s uncertainty about system state increases with time since his last move.

Moves are “stealthy”

- In practice, compromise is often undetected...
- In FlipIt,
  players do *not* immediately know when the other player makes a move!
  (Very unusual in game theory literature!)
- Player’s uncertainty about system state increases with time since his last move.
- A move may *take control* (“flip”) or *have no effect* (“flop”).

Moves are “stealthy”

- In practice, compromise is often undetected...
- In FLIPIT,
  players do *not* immediately know when the other player makes a move!
  (Very unusual in game theory literature!)
- Player’s uncertainty about system state increases with time since his last move.
- A move may *take control* (“flip”) or *have no effect* (“flop”).
- Uncertainty means flops are unavoidable.

Moves may be informative

- A player learns the state of the system *only* when he moves.

Moves may be informative

- A player learns the state of the system *only* when he moves.
- In basic `FLIPIT`, each move has feedback that reveals all previous moves.

Moves may be informative

- A player learns the state of the system *only* when he moves.
- In basic FLIPIT, each move has feedback that reveals all previous moves.
- In variants, move reveals only current state, or time since other player last moved...

Cost of moves and gains for being in control

- Moves aren't for free!

Cost of moves and gains for being in control

- Moves aren't for free!
- Player i pays k_i points per move:
  Defender pays k_0, Attacker pays k_1

Cost of moves and gains for being in control

- Moves aren't for free!
- Player i pays k_i points per move:
  Defender pays k_0, Attacker pays k_1

- Being in control yields gain!

Cost of moves and gains for being in control

- Moves aren't for free!
- Player i pays k_i points per move:
  Defender pays k_0, Attacker pays k_1

- Being in control yields gain!
- Player earns one point for each second he is in control.

How well are you playing? (Notation)

- Let $N_i(t)$ denote number moves by player i up to time t. His average rate of play is

$$\alpha_i(t) = N_i(t)/t \ .$$

How well are you playing? (Notation)

- Let $N_i(t)$ denote number moves by player i up to time t. His average rate of play is

$$\alpha_i(t) = N_i(t)/t \ .$$

- Let $G_i(t)$ denote the number of seconds player i is in control, up to time t.
  His rate of gain up to time t as

$$\gamma_i(t) = G_i(t)/t \ .$$

How well are you playing? (Notation)

- Score (net benefit) $B_i(t)$ up to time t is TimeInControl - CostOfMoves:

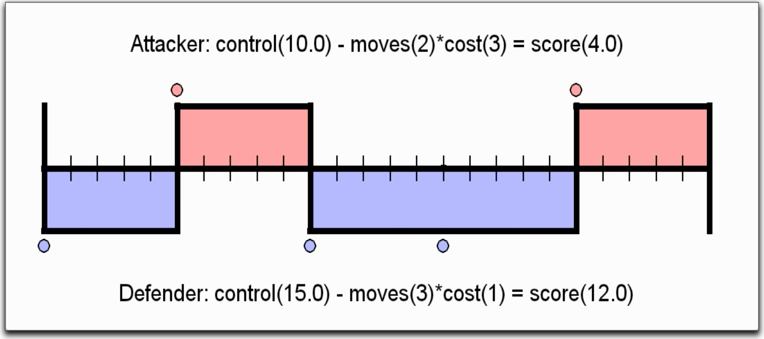
$$B_i(t) = G_i(t) - k_i \cdot N_i(t)$$

- Benefit rate is

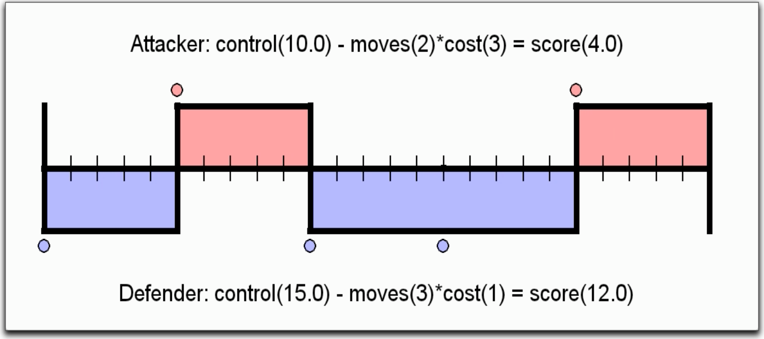
$$\begin{aligned} \beta_i(t) &= B_i(t)/t \\ &= \gamma_i(t) - k_i \cdot \alpha_i(t) \end{aligned}$$

- Player wishes to maximize $\beta_i = \lim_{t \to \infty} \beta_i(t)$.

Movie of FLIPIT Game – Global View



Movie of FLIPIT Game – Defender View



How to play well?

Non-Adaptive Play

Non-adaptive strategies

- A *non-adaptive strategy* plays on blindly, independent of other player's moves.

Non-adaptive strategies

- A *non-adaptive strategy* plays on blindly, independent of other player's moves.
- In principle, a non-adaptive player can pre-compute his entire (infinite!) list of moves before the game starts.

Non-adaptive strategies

- A *non-adaptive strategy* plays on blindly, independent of other player's moves.
- In principle, a non-adaptive player can pre-compute his entire (infinite!) list of moves before the game starts.
- Some interesting non-adaptive strategies:

Non-adaptive strategies

- A *non-adaptive strategy* plays on blindly, independent of other player's moves.
- In principle, a non-adaptive player can pre-compute his entire (infinite!) list of moves before the game starts.
- Some interesting non-adaptive strategies:
  - *Periodic* play

Non-adaptive strategies

- A *non-adaptive strategy* plays on blindly, independent of other player's moves.
- In principle, a non-adaptive player can pre-compute his entire (infinite!) list of moves before the game starts.
- Some interesting non-adaptive strategies:
  - *Periodic* play
  - *Exponential* (memoryless) play

Non-adaptive strategies

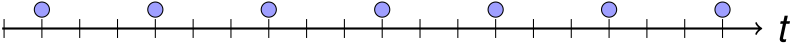
- A *non-adaptive strategy* plays on blindly, independent of other player's moves.
- In principle, a non-adaptive player can pre-compute his entire (infinite!) list of moves before the game starts.
- Some interesting non-adaptive strategies:
  - *Periodic* play
  - *Exponential* (memoryless) play
  - *Renewal* strategies: iid intermove times

Periodic play

Player *i* may play *periodically*
with rate α_i and period $1/\alpha_i$

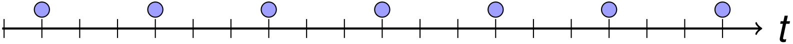
Periodic play

Player i may play *periodically*
with rate α_i and period $1/\alpha_i$
E.g. for $\alpha_0 = 1/3$, we might have:



Periodic play

Player *i* may play *periodically*
with rate α_i and period $1/\alpha_i$
E.g. for $\alpha_0 = 1/3$, we might have:



It is convenient to assume that periodic play involves miniscule amounts of jitter or drift; play is effectively periodic but will drift out of phase with truly periodic.

Adaptive play against a periodic opponent

An *adaptive* Attacker can easily learn the period and phase of a periodic Defender, so that periodic play is useless against an adaptive opponent, unless it is very fast.

Examples:

- a sentry make his regular rounds
- 90-day password reset

Periodic Attacker

Theorem

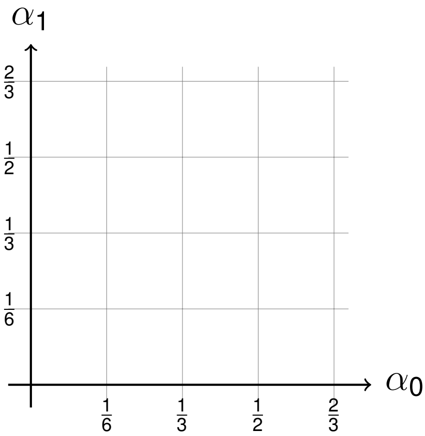
If Attacker moves periodically at rate α_1 (and period $1/\alpha_1$, with unknown phase), then optimum non-adaptive Defender strategy is

- *if $\alpha_1 > 1/2k_0$, don't play(!),*
- *if $\alpha_1 = 1/2k_0$, play periodically at any rate α_0,* $0 \leq \alpha_0 \leq 1/2k_0$,
- *if $\alpha_1 < 1/2k_0$, play periodically at rate*

$$\alpha_0 = \sqrt{\frac{\alpha_1}{2k_0}} > \alpha_1$$

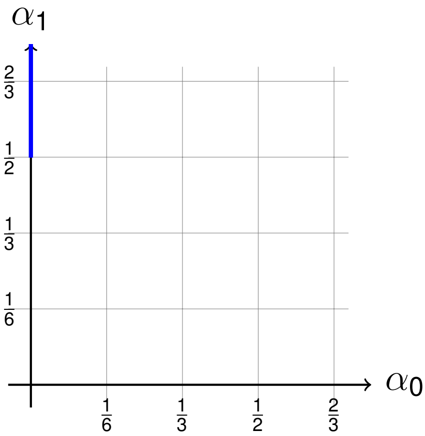
Graph for Periodic Attacker and Periodic Defender

$(k_0 = 1, k_1 = 1.5)$



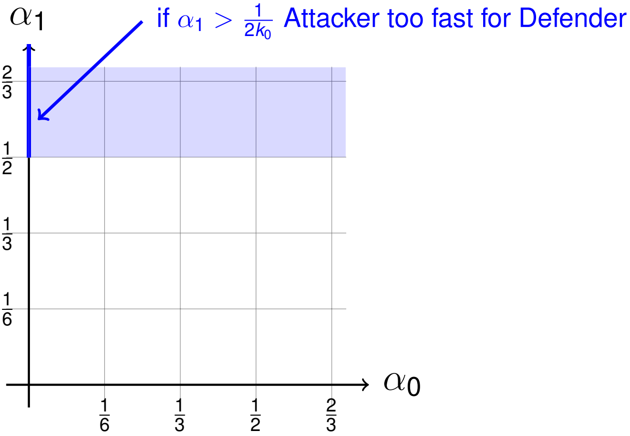
Graph for Periodic Attacker and Periodic Defender

$(k_0 = 1, k_1 = 1.5)$



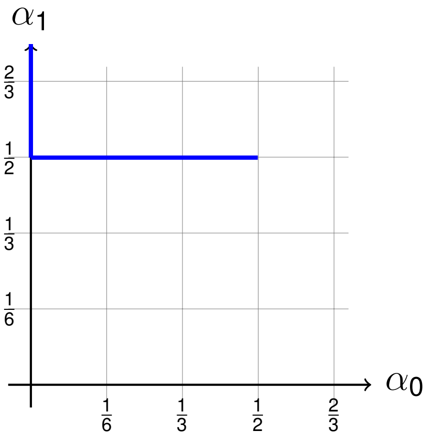
Graph for Periodic Attacker and Periodic Defender

$(k_0 = 1, k_1 = 1.5)$



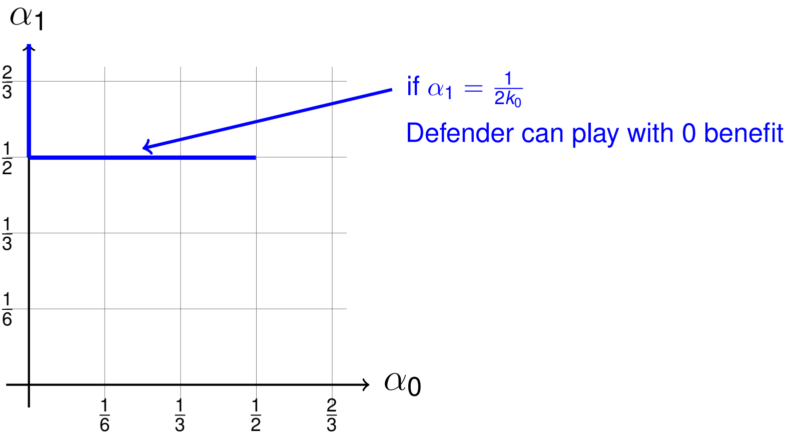
Graph for Periodic Attacker and Periodic Defender

$(k_0 = 1, k_1 = 1.5)$



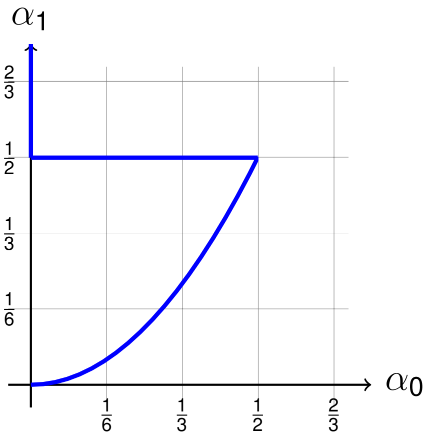
Graph for Periodic Attacker and Periodic Defender

$(k_0 = 1, k_1 = 1.5)$



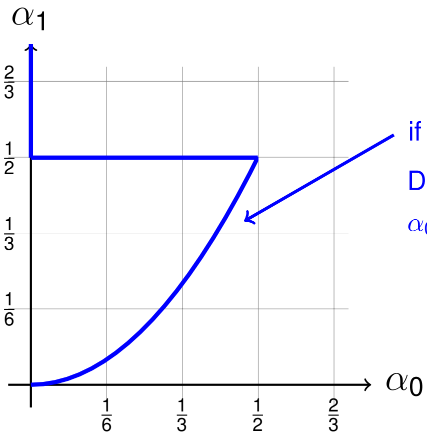
Graph for Periodic Attacker and Periodic Defender

$(k_0 = 1, k_1 = 1.5)$



Graph for Periodic Attacker and Periodic Defender

$(k_0 = 1, k_1 = 1.5)$



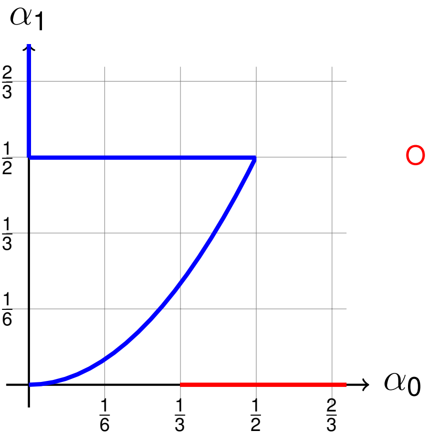
if $\alpha_1 < \frac{1}{2k_0}$

Defender maximizes benefit with

$\alpha_0 = \sqrt{\frac{\alpha_1}{2k_0}}$

Graph for Periodic Attacker and Periodic Defender

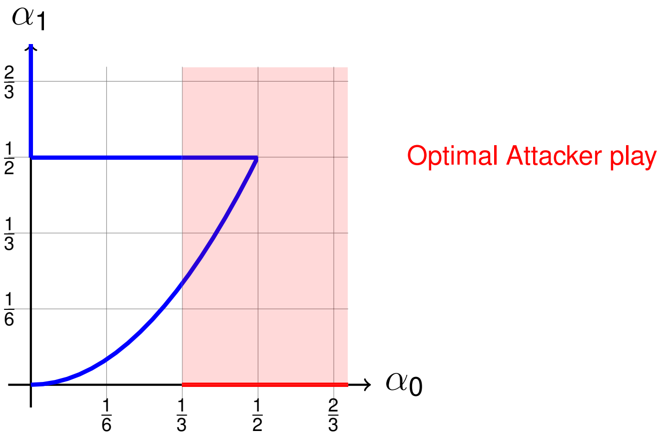
$(k_0 = 1, k_1 = 1.5)$



Optimal Attacker play

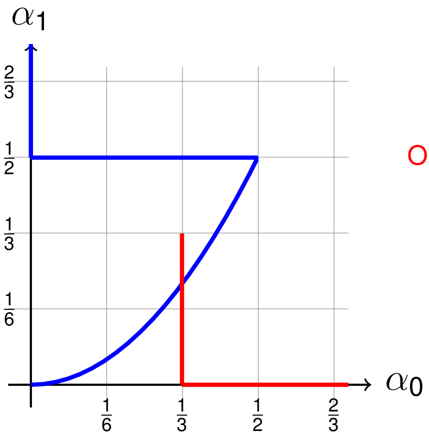
Graph for Periodic Attacker and Periodic Defender

$(k_0 = 1, k_1 = 1.5)$



Graph for Periodic Attacker and Periodic Defender

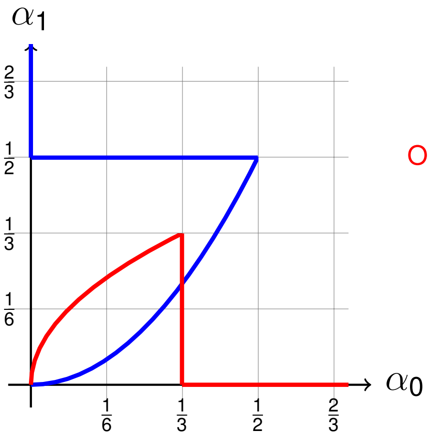
$(k_0 = 1, k_1 = 1.5)$



Optimal Attacker play

Graph for Periodic Attacker and Periodic Defender

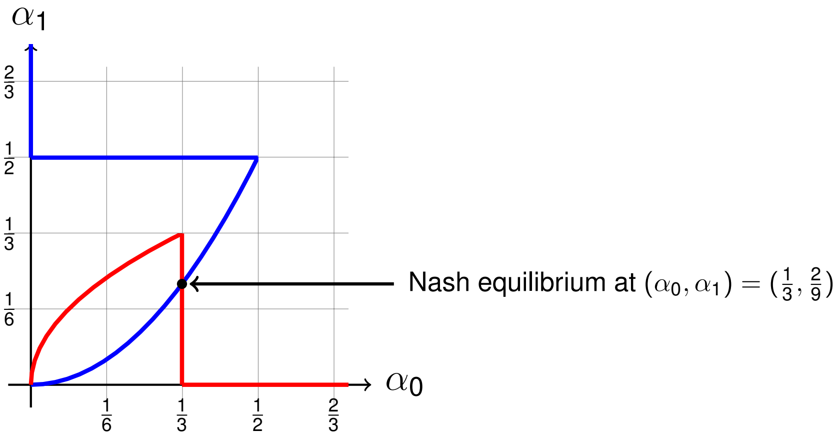
$(k_0 = 1, k_1 = 1.5)$



Optimal Attacker play

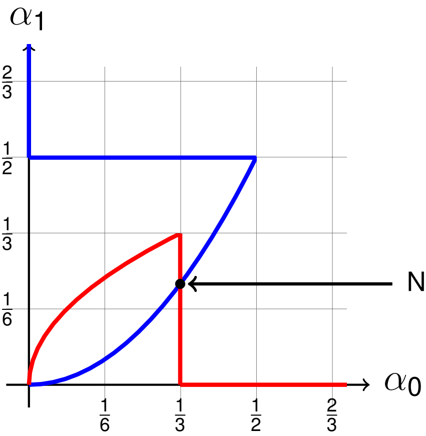
Graph for Periodic Attacker and Periodic Defender

$(k_0 = 1, k_1 = 1.5)$



Graph for Periodic Attacker and Periodic Defender

$(k_0 = 1, k_1 = 1.5)$



Nash equilibrium at $(\alpha_0, \alpha_1) = (\frac{1}{3}, \frac{2}{9})$
$(\gamma_0, \gamma_1) = (\frac{2}{3}, \frac{1}{3})$
$(\beta_0, \beta_1) = (\frac{1}{3}, 0)$

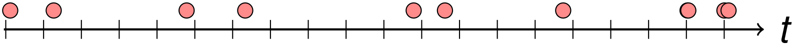
Exponential Attacker

If Attacker plays exponentially with rate α_1, then his moves form a memoryless Poisson process; he plays independently in each interval of time of size *dt* with probability $\alpha_1\, dt$

Probability that intermove delay is at most *x* is

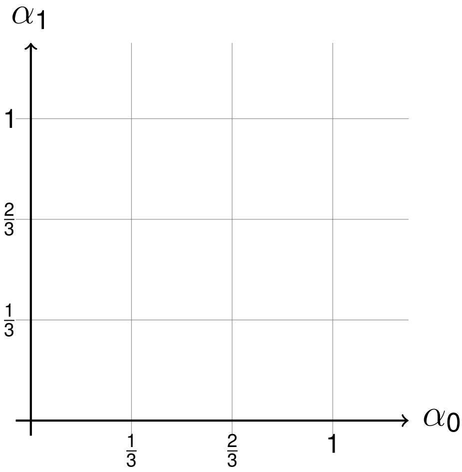
$$1 - e^{-\alpha_1 x}$$

For $\alpha_1 = 0.5$, we might have:



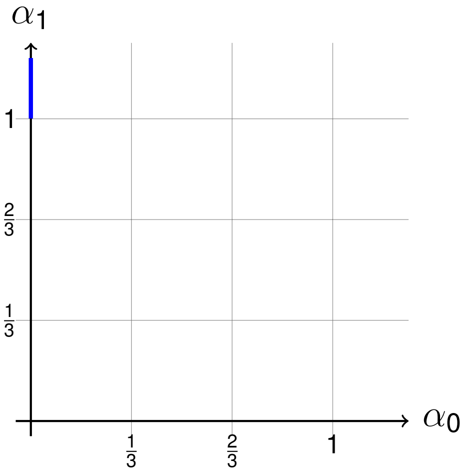
Graph for Exponential Attacker and Defender)

($k_0 = 1, k_1 = 1.5$)



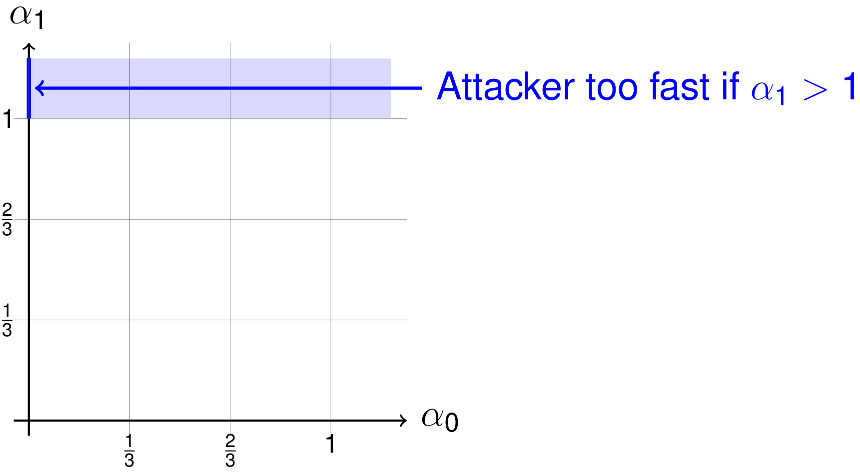
Graph for Exponential Attacker and Defender)

($k_0 = 1, k_1 = 1.5$)



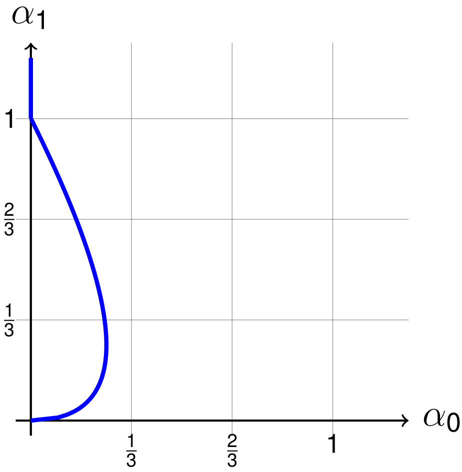
Graph for Exponential Attacker and Defender)

$(k_0 = 1, k_1 = 1.5)$



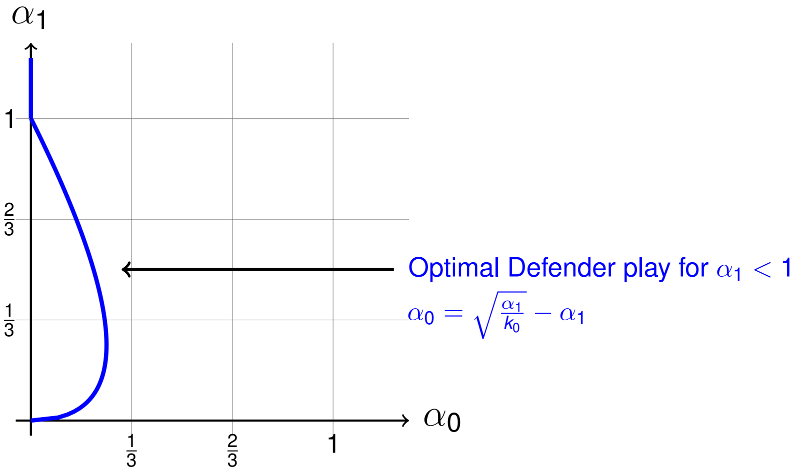
Graph for Exponential Attacker and Defender)

$(k_0 = 1, k_1 = 1.5)$



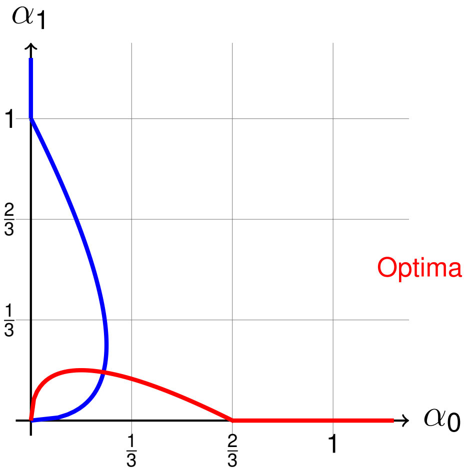
Graph for Exponential Attacker and Defender)

$(k_0 = 1, k_1 = 1.5)$



Graph for Exponential Attacker and Defender)

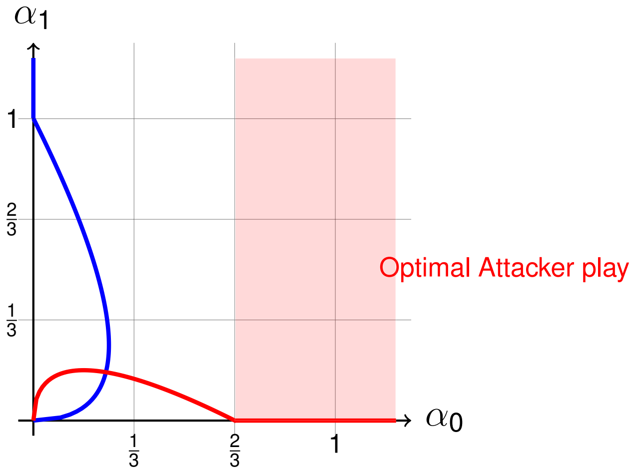
$(k_0 = 1, k_1 = 1.5)$



Optimal Attacker play

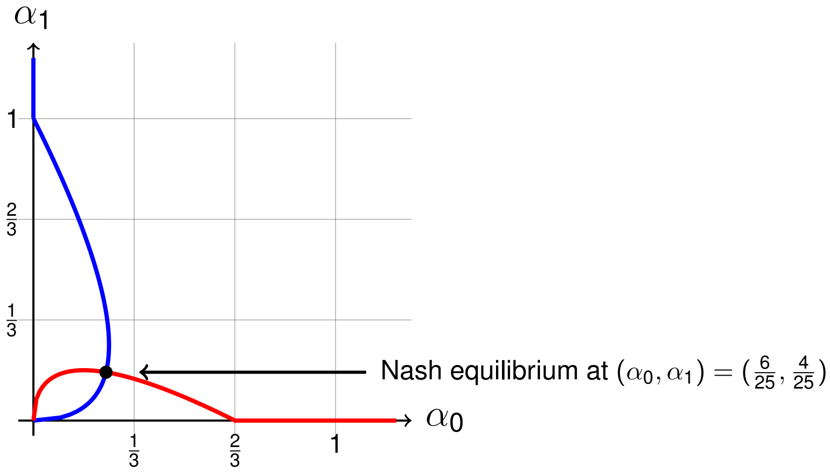
Graph for Exponential Attacker and Defender)

$(k_0 = 1, k_1 = 1.5)$



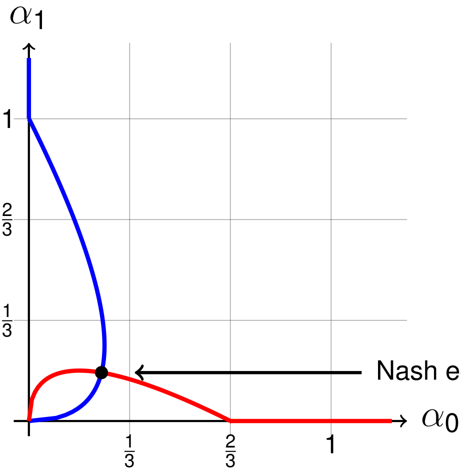
Graph for Exponential Attacker and Defender)

$(k_0 = 1, k_1 = 1.5)$



Graph for Exponential Attacker and Defender)

$(k_0 = 1, k_1 = 1.5)$



Nash equilibrium at $(\alpha_0, \alpha_1) = (\frac{6}{25}, \frac{4}{25})$

$(\gamma_0, \gamma_1) = (\frac{3}{5}, \frac{2}{5})$

$(\beta_0, \beta_1) = (\frac{9}{25}, \frac{6}{25})$

Renewal Strategies

A *renewal strategy* is one with iid intermove delays for player *i*'s moves:

$$\Pr(\text{delay} \leq x) = F_i(x)$$

for some distribution F_i.

Renewal strategies form a very large class of (non-adaptive) strategies; periodic, exponential, etc. are special cases...
Origin of term: player's moves form a *renewal process*.

Optimal (renewal) play against a renewal strategy.

One of our major results is the following:

Theorem

The optimal renewal strategy against any renewal strategy is either periodic or not playing.

Proof notes

Average time between buses

$$\neq$$

Average waiting time for a bus

Proof notes

Average time between buses

$$\neq$$

Average waiting time for a bus

Proof considers *size-biased* interval sizes...

Proof notes

Average time between buses
$$\neq$$
Average waiting time for a bus

Proof considers *size-biased* interval sizes...

Note that a periodic strategy minimizes variance of interval sizes, and thus minimizes size-biased interval size.

Adaptive Play

Adaptive Strategies

- Periodic strategy not very effective against *adaptive* Attacker, who can learn to move just after each Defender move.

Adaptive Strategies

- Periodic strategy not very effective against *adaptive* Attacker, who can learn to move just after each Defender move.
- FLIPIT with adaptive strategies can be complicated – generalizes iterated Prisoner's Dilemma—e.g. for periodic play:

Adaptive Strategies

- Periodic strategy not very effective against *adaptive* Attacker, who can learn to move just after each Defender move.
- FLIPIT with adaptive strategies can be complicated – generalizes iterated Prisoner's Dilemma—e.g. for periodic play:

	slow($\alpha_1 = 0.1$)	fast($\alpha_1 = 0.2$)
slow($\alpha_0 = 0.1$)	0.40,0.40	-0.10,0.55
fast($\alpha_0 = 0.2$)	0.55,-0.10	0.30,0.30

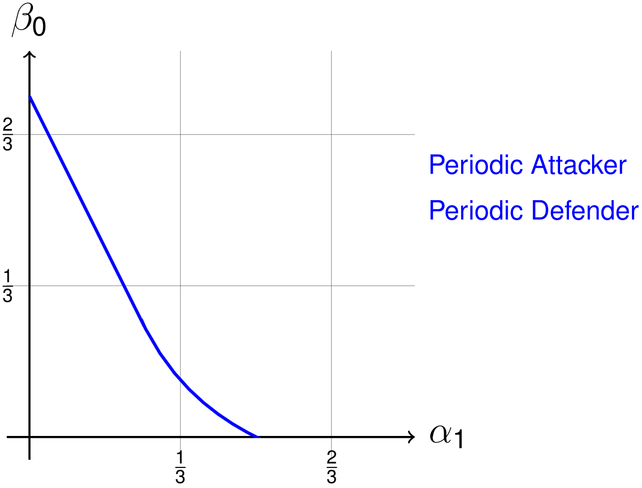
Exponential works well even against adaptive strategies

Theorem

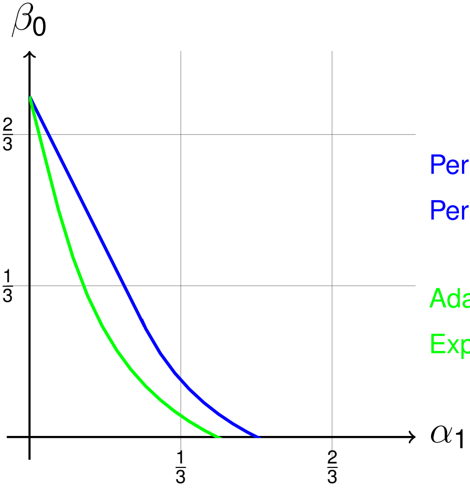
The optimal strategy (of any sort, even adaptive) against an exponential strategy is either periodic or not playing.

Defender can always play exponential strategy against a potentially adaptive Attacker; Attacker can't then do better than playing periodically (or not playing).

Defender's ($\alpha_0 = 0.25$) net benefit β_0 against optimal (periodic) Attacker (α_1 *variable*)



Defender's ($\alpha_0 = 0.25$) net benefit β_0 against optimal (adaptive) Attacker (α_1 *variable*)



Periodic Attacker

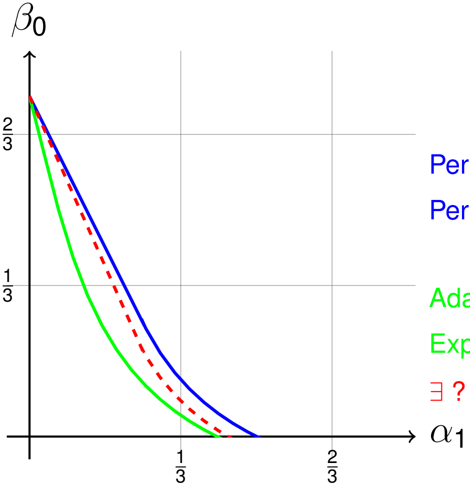
Periodic Defender

$\Downarrow$

Adaptive Attacker

Exponential Defender

Defender's ($\alpha_0 = 0.25$) net benefit β_0 against optimal (adaptive) Attacker (α_1 *variable*)



Periodic Attacker

Periodic Defender

$\Downarrow$

Adaptive Attacker

Exponential Defender

$\exists$? Better Defender ?

Lessons and Open Questions

Lessons

- Be prepared to deal with continual repeated failure (loss of control).

Lessons

- Be prepared to deal with continual repeated failure (loss of control).
- Play fast! Aim to make opponent drop out! (Agility!)

Lessons

- Be prepared to deal with continual repeated failure (loss of control).
- Play fast! Aim to make opponent drop out! (Agility!)
- Arrange game so that your moves cost much less than your opponent's!
  (Cheap to refresh passwords or keys, easy to reset system to pristine state (as with a virtual machine))

Open question 1

Conjecture: The optimal non-adaptive strategy against a renewal strategy is periodic.

(We only proved that optimal renewal strategy is periodic.)

Open question 2

What is “optimal” renewal strategy against an adaptive rate-limited Attacker?
(e.g. $N_1(t)/t \leq \alpha_1$ for all t)?

Open question 2

What is "optimal" renewal strategy against an adaptive rate-limited Attacker?
(e.g. $N_1(t)/t \leq \alpha_1$ for all t)?

That is, how to balance trade-off between periodic play, which has low-variance intervals but is predictable, and exponential, which has high-variance intervals but is very unpredictable?

Perhaps using gamma-distributed intervals or delayed exponentials?

Open question 3

Are there information-theoretic bounds on how well a rate-limited Attacker can do against a fixed renewal strategy by Defender?

Open question 4

What learning theory algorithms yield adaptive strategies provably optimal against renewal strategies?

Open questions 5, 6, 7, ...

5 Multi-player FLIPIT

6 Other feedback models (e.g. add low-cost "check")

7 How to structure PKI when any party (including CA's) may get "hacked" at any time?

... ...

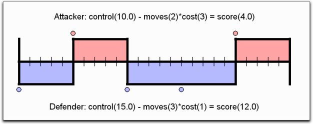
Online version of FLIPIT

More information on FLIPIT, including an online interactive version of the game, will be available in the next few weeks at:

`www.rsa.com/flipit`

Enjoy!

The End


Attacker: control(10.0) - moves(2)*cost(3) = score(4.0)
Defender: control(15.0) - moves(3)*cost(1) = score(12.0)