Distinguished Speaker Seminar
27 October 2015
Concordia Institute for Information Systems Engineering, Montreal

Password Expiration Policies: Analyzing the Security Benefits

Paul C. Van Oorschot
School of Computer Science
Carleton University, Ottawa, Canada

These slides complement the paper:

Quantifying the security advantage
of password expiration policies

Sonia Chiasson, P.C. van Oorschot

[DOI: 10.1007/s10623-015-0071-9]

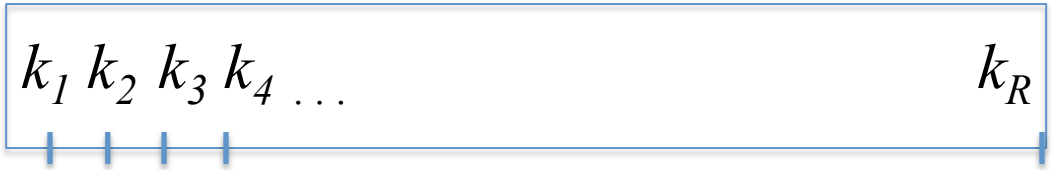
Designs, Codes & Cryptography, 77(2):401-408, 2015
Special issue in memory of Scott A. Vanstone

- Password aging policies
- Benefits:
  - Qualitative
  - Quantitative benefit from password change?

Q: You change your password continuously,
as quickly as system interfaces allow.
Does this prevent successful guessing attacks?

Starting Model /1

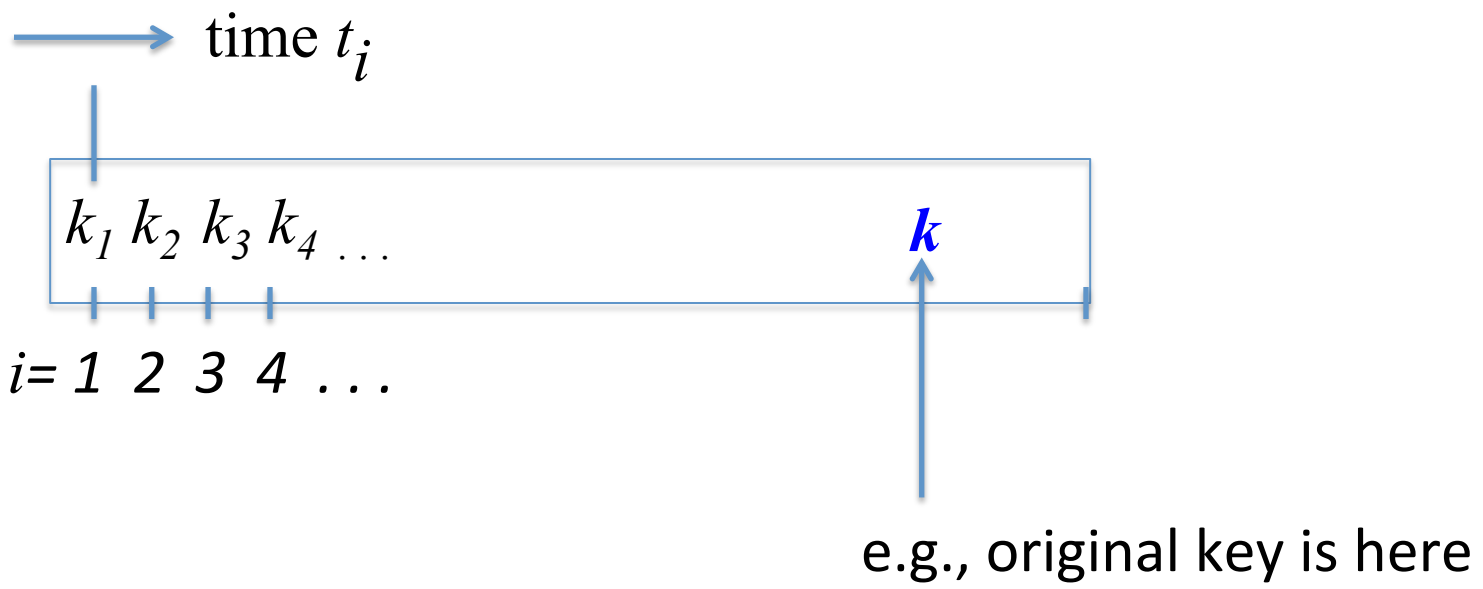
- crypto key search, randomly chosen key
- assume exhaustive attack on account
  - online guessing (for simplicity)
  - deterministic finite search, R elements (certain success if no change)
  - attacker knows policy period length, not time of password changes
  - on reaching end of search space, restart (possibly different order)



$R = 2^r$

Starting Model /2

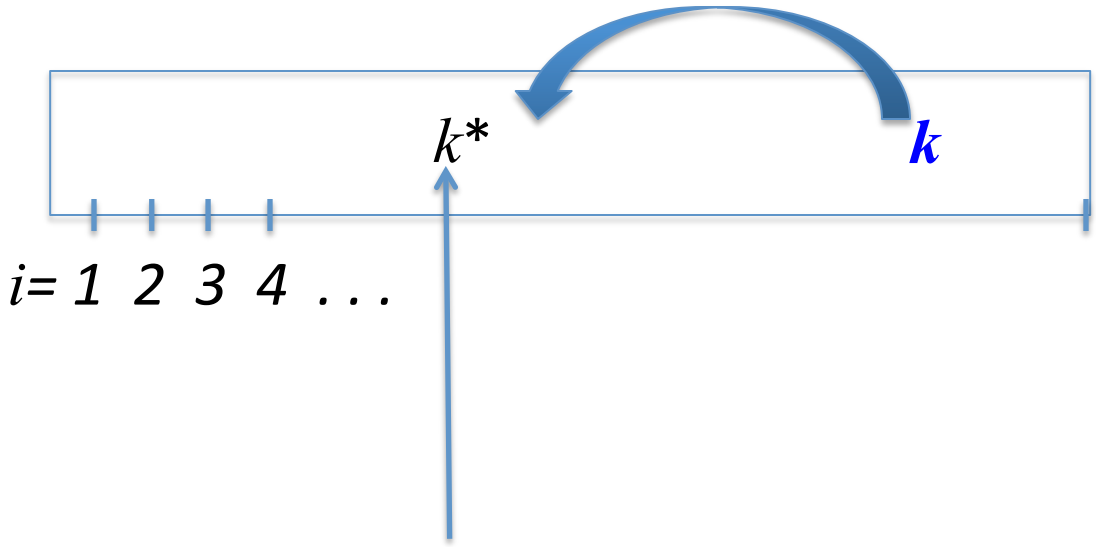
- user's key is $k \in \{k_1, k_2, \ldots, k_R\}$
- attacker guesses key k_i at time t_i



Starting Model /3

- k changed to k^* at $t_{u+1} \in \{t_2, \ldots, t_R\}$

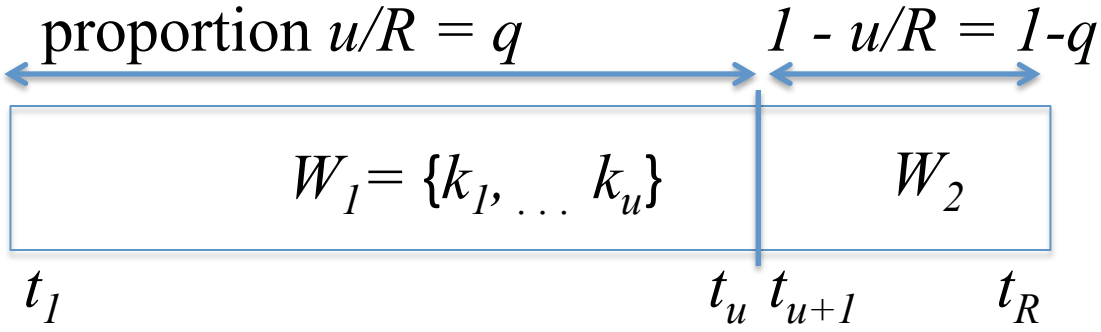
- Q: what security advantage results
  e.g., delta in prob(successful guess over R guesses)?



e.g., new key is here

Base Analysis ($T \leq P$) /1

- success means finding either k or k^* while it is active
- assume one change in a fixed policy period of length P
- allow time T (exhaustion time) = time to make R guesses



Case	Events	Result	Probability
1	$k \in W_1, k^* \in W_2$	success	$q_1 = (q)(1-q)$
2	$k \in W_1, k^* \notin W_2$	success	$q_2 = (q)(q)$
3	$k \notin W_1, k^* \in W_2$	success	$q_3 = (1-q)(1-q)$
4	$k \notin W_1, k^* \notin W_2$	failure	$q_4 = (1-q)(q)$

Base Analysis ($T \leq P$) /2

- prob(attack success) is $p_s = q_1 + q_2 + q_3 = 1 - q + q^2$

$$p_f = q - q^2$$

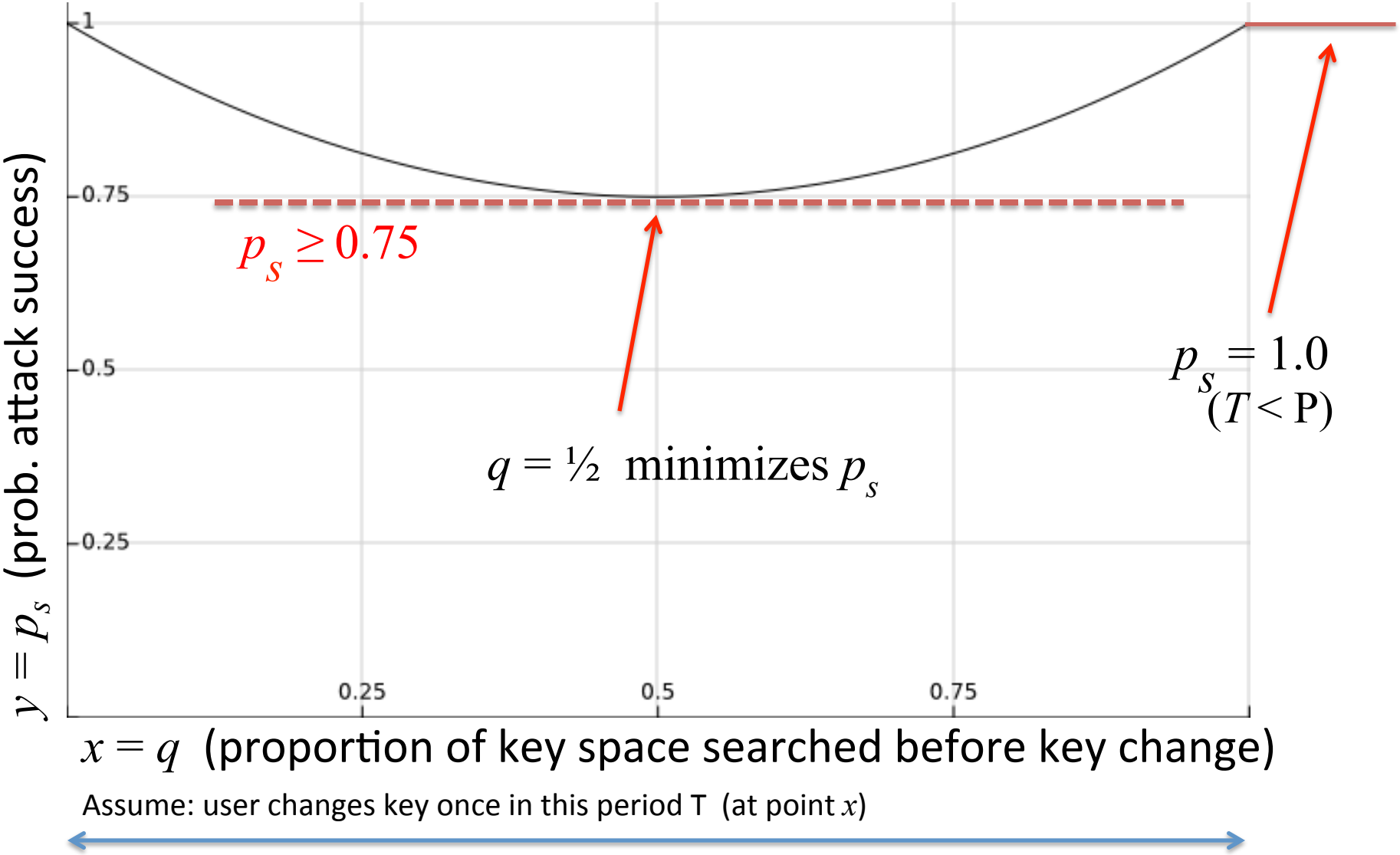
- min/max at $q = 1/2$

$$p_f(q = 1/2) = 1/2 - 1/4 = 1/4$$

$$p_s(q = 1/2) = 3/4$$

Result	Probability
success	$q_1 = (q)(1-q)$
success	$q_2 = (q)(q)$
success	$q_3 = (1-q)(1-q)$
failure	$q_4 = (1-q)(q)$

Probability of attacker success ($T \leq P$; single search T)



Assume: user changes key once in this period T (at point x)

Case $T \le P$ [cont'd]

- $p_s \ge 0.75$ for single exhaustive search T

And if search fails, then what?

. . . attacker repeats search (possibly distinct search order)

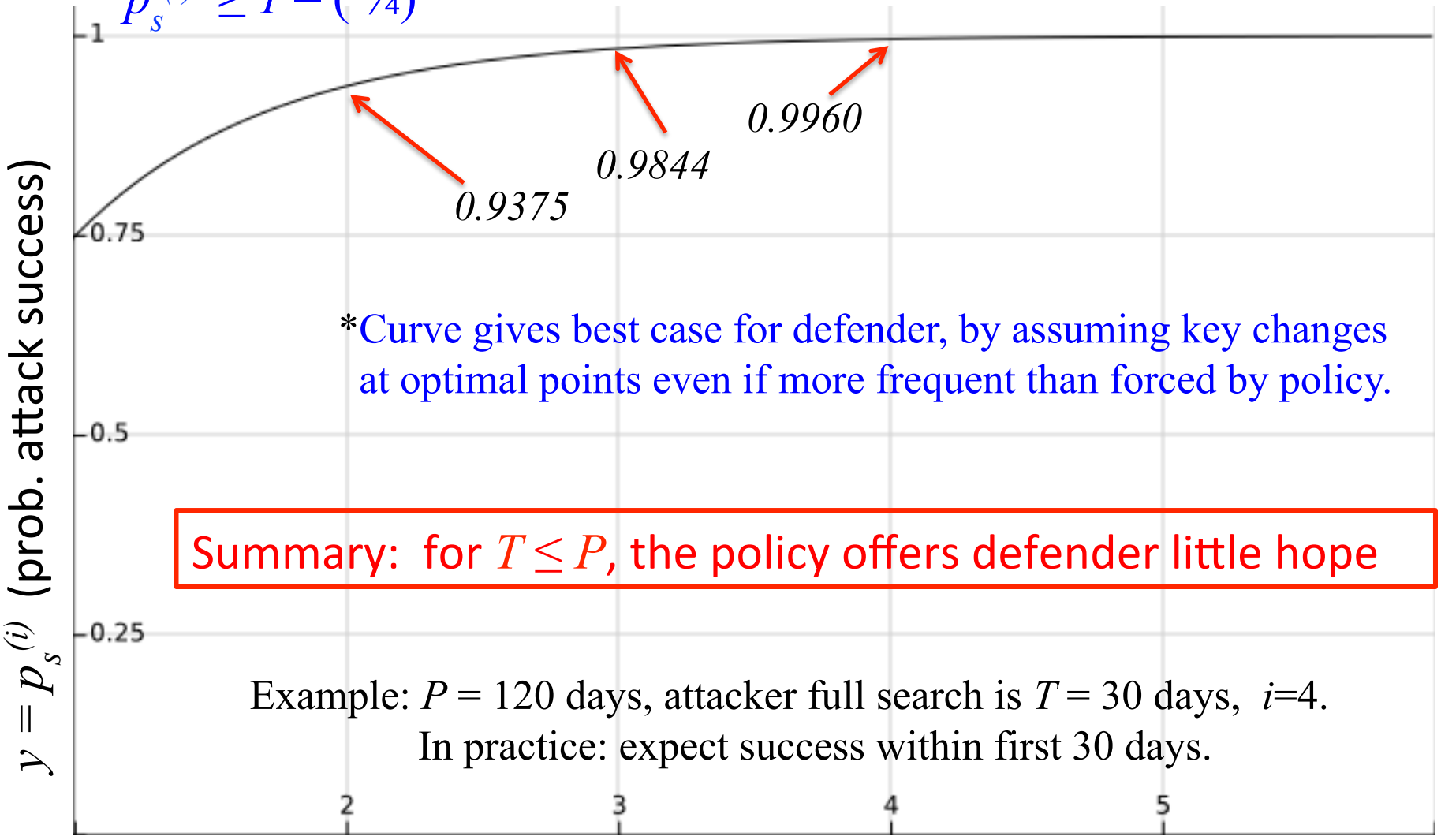
prob(success over i search periods T, i.e., time $i \cdot T$) is

$$p_s^{(i)} = 1 - \text{ prob.(failure on all } i \text{ searches)}$$

$$\ge 1 - (\tfrac{1}{4})^i$$

Probability of attack success over i repeated searches T, $T \leq P$*

$p_s^{(i)} \geq 1 - (1/4)^i$



*Curve gives best case for defender, by assuming key changes at optimal points even if more frequent than forced by policy.

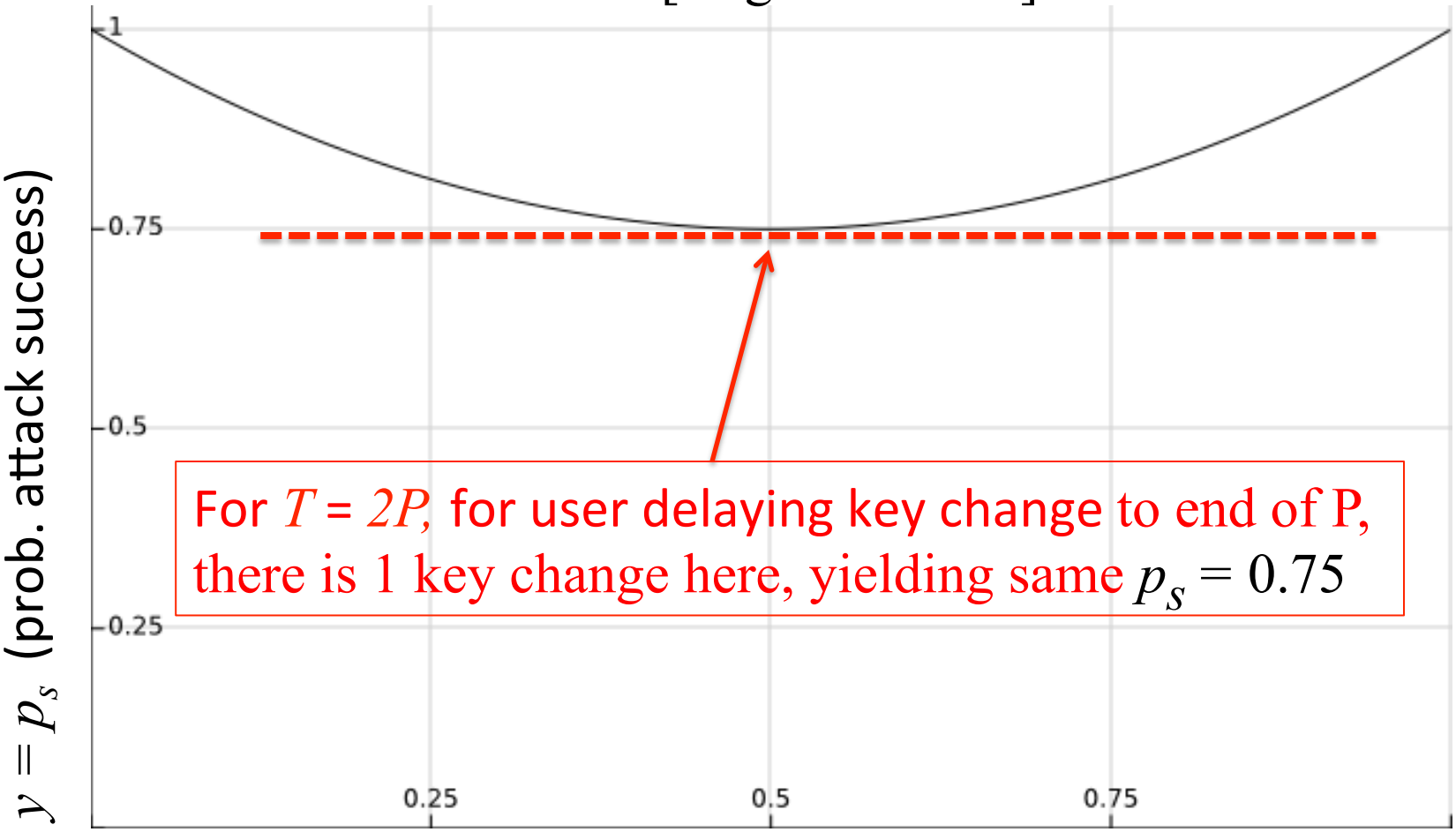
Summary: for $T \leq P$, the policy offers defender little hope

Example: P = 120 days, attacker full search is T = 30 days, i=4.
In practice: expect success within first 30 days.

$x = i$ (number of repeated exhaustive searches each over time T)

Base analysis also informs re: expected behaviour for

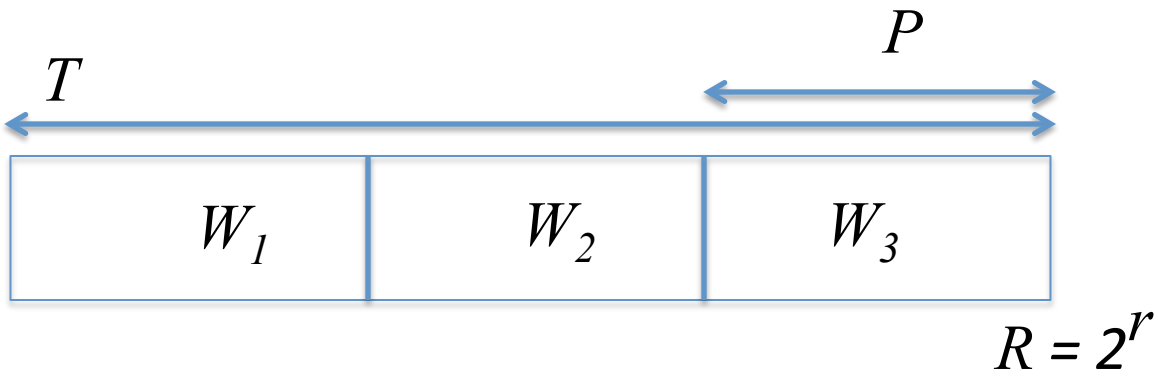
$T = 2P$ [single search T]



General case $T = t \cdot P$ $[t \geq 2]$

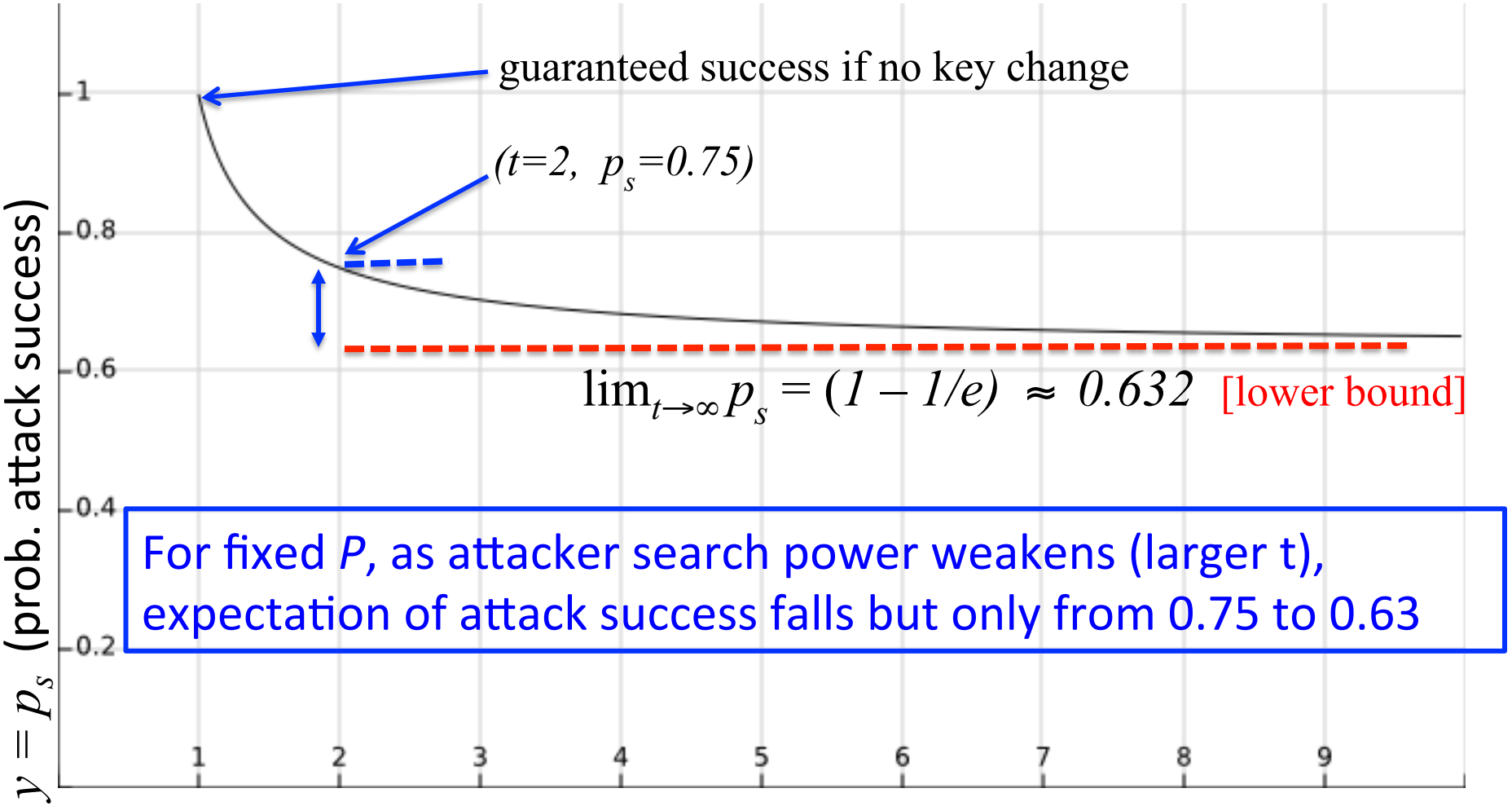
- each segment W_i houses $1/t$ of the R keys
  - consider $t = 3$ for concreteness, to help in reasoning
- success = guessing at least 1 of the $k^{(i)}$ while active in W_i

  … so (per period T) attack fails iff for all $1 \leq i \leq t$: $k^{(i)} \notin W_i$
- thus $p_f = (1 - 1/t)^t$ and $p_s = 1 - (1 - 1/t)^t$



$$p_s = 1 - (1 - 1/t)^t$$

[$T = t \cdot P$; single search T]



$x = t$ ($T = t \cdot P$ is exhaustive search time; P = expiration period)

So for case $T = t \bullet P$ (single exhaustive search period T)

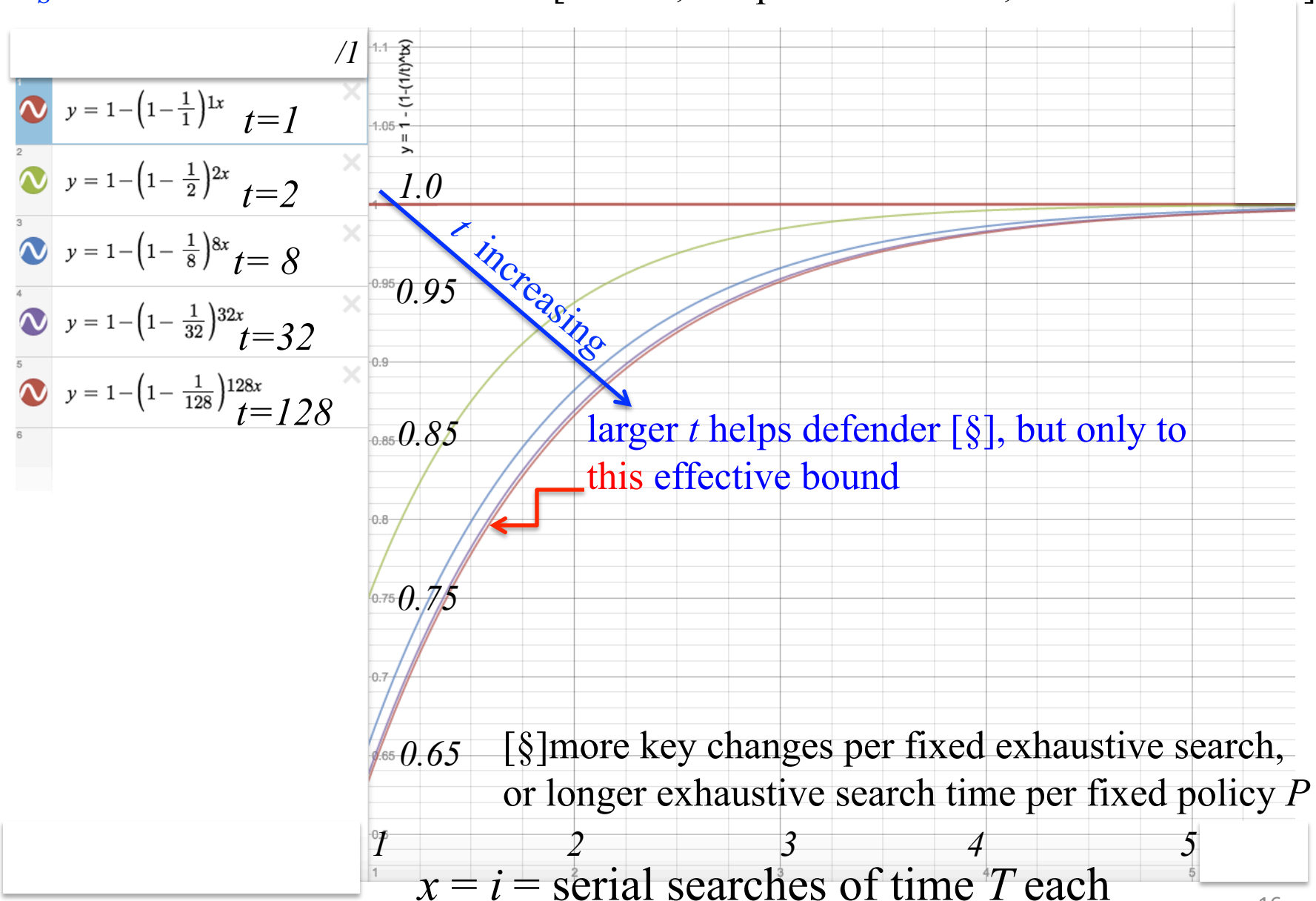
$$p_s = 1 - (1 - 1/t)^t$$

Next, consider $i > 1$ search periods T, i.e., attack time $iT = itP$

Corresponding result is $p_s^{(t,i)} = 1 - (1 - 1/t)^{ti}$

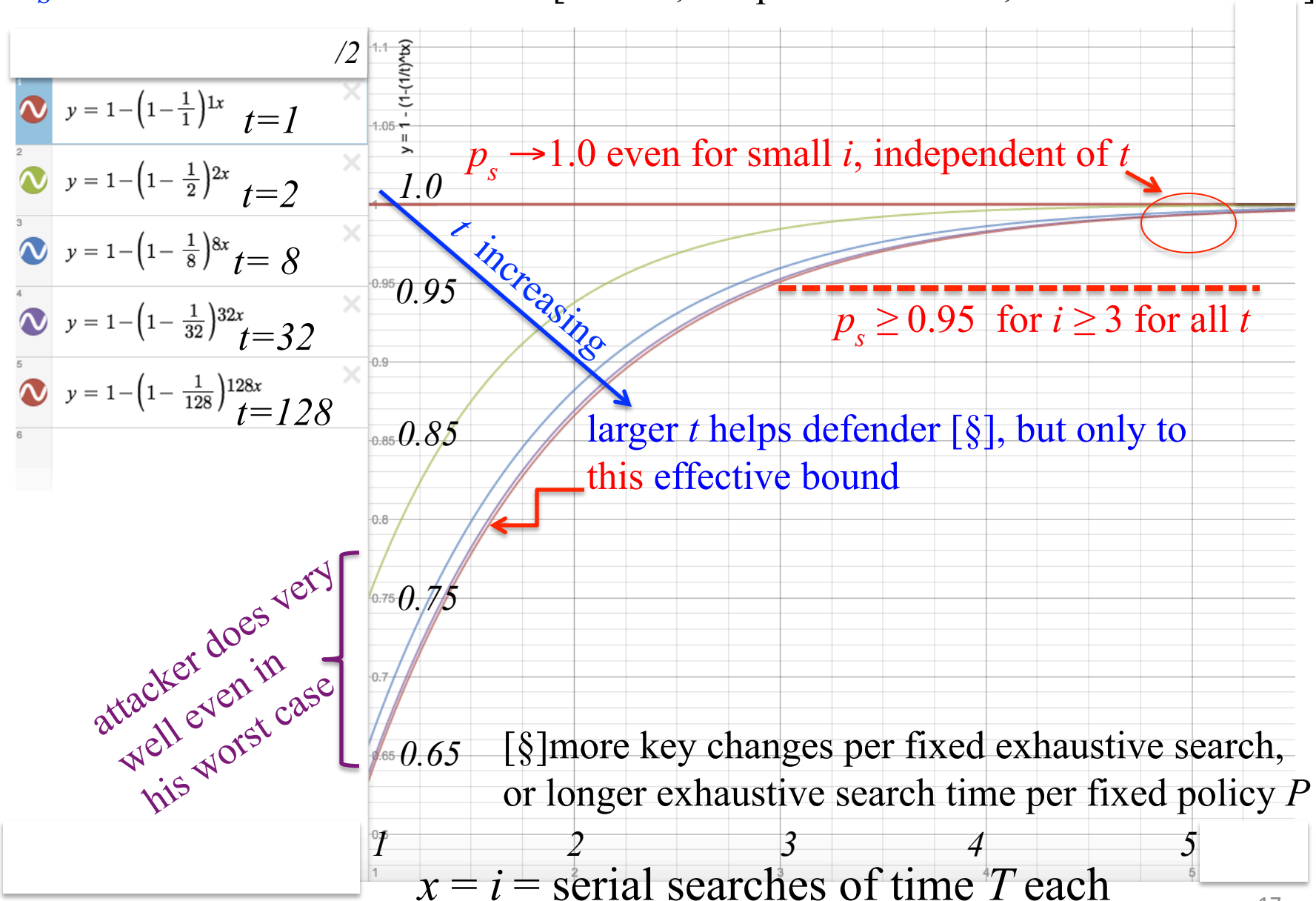
$$p_s^{(t,i)} = 1 - (1 - 1/t)^{ti}$$

[$T = t \cdot P$, i repeated searches, total time $iT = itP$]



$$p_s^{(t,i)} = 1 - (1 - 1/t)^{ti}$$

[$T = t \cdot P$, i repeated searches, total time $iT = itP$]



Remark: T >> P

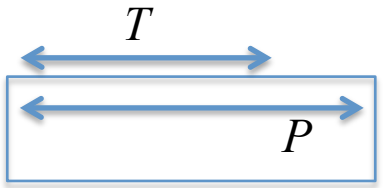
- case: key changes very often before full exhaustive search
- more important than number of changes is time for full search
- prob(attack success) high even on key change after every guess
  - recall $\lim_{t \to \infty} p_s$

Remark: Offline attacks

- original goal of password expiration policy:
  limit risk (1 year) of guessing attack compromise to . . . 1 in 10^6
- unattainable today (offline, no iterated hashing):
  modern processing resources easily allow 7-10 billion guesses/s
- ex: 8-char totally random password, from 93-symbol alphabet,
  - $93^8 = 2^{52.3}$ elements
  - searchable in 9.2 days (with quite modest resources)
  - 1 in 1 million chance: requires password change every 800ms
- conclude: essentially no protection against offline guessing

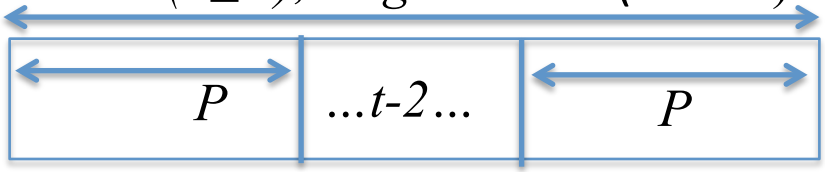
Partial Summary

$T \le P$, *single* search



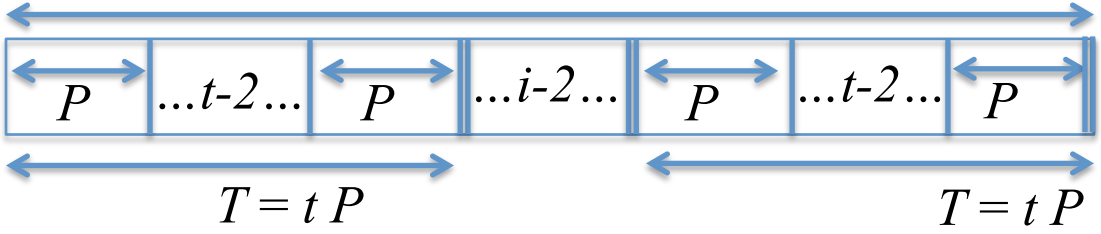
Single key change at end of T: $p_s = 1.0$
Optimal defense (change at $T/2$): $p_s \ge 0.75$
and $p_s \rightarrow 1.0$ rapidly on repeated searches

$T = tP$ $(t \ge 3)$, *single* search (time T)



Lower bound on p_s drops,
but only to: $p_s \ge 0.632$

$iT = i\,t\,P$, in total i serial exhaustive searches



$p_s \rightarrow 1.0$ even for small i,
independent of t

Sure, but how about user passwords (vs. equi-probable crypto keys)?

Relating to User-Chosen Passwords & Aging

For user-chosen passwords: analysis not as clean

(a) password length variation

(b) skewed distributions

But (a) is easy to model approximately.
And for (b), insight from large empirical datasets (next slide)

Real attackers . . .

- optimize by guessing in (estimated) probability order
- *quit early* (abandon long tail of key space)
- offline attack: "almost" full searches within reach as before
- same p_s over full T as for equi-probable keys, but shorter expected time to success due to skew: greatly helps online attackers

Password Aging: Empirical Studies and Skew

Bonneau (Oakland 2012; natural dataset of 70m)

- online guessing trying most-popular passwords on each of large # of accounts (e.g., 10/account), yields ≈ 1% of passwords
- optimal attacker, massive search: gets 50% after 1m guesses/acct

Weir (CCS 2010; analysis, including dataset of 32m)

- most popular 50,000 items from training sub-list of 5m: covers over 25% other sub-lists (for length-7 or more)

Thus for user-chosen passwords, results are even worse for defender than results from idealized crypto model.

Does password expiration stop guessing attacks?

No. If passwords are guessable*, then they are guessable

- playing "hide-and-seek"

If attack vector is NOT guessing, then expiration can temporarily terminate ongoing access. But . . .

- doesn't prevent continued access by consequent backdoors
- doesn't undo damage upon original access (barn door)
- doesn't stop attack vectors which may *re-execute*: persistent client-malware, persistent network interception

*[enough guesses can be tested to pass relevant threshold of success probability]

So in the end: what help do aging policies provide?

1. may temporarily disrupt ongoing/post-compromise access:
   - for case of "delegated" access or "group-shared" password
   - for attacker capturing passwords to sell, if password changed before account access by purchaser [but see Zhang]
2. forces offline attacker [relatively rare] to acquire new hash file

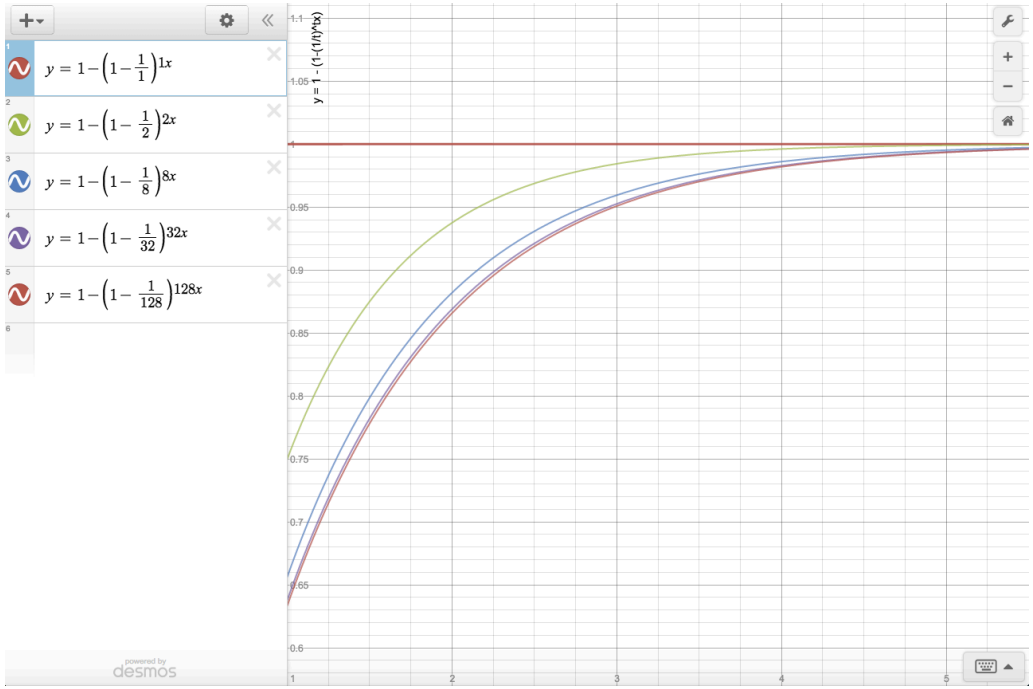
But little help vs. guessing attack … and yet more cons in all cases:

- heavy usability impact
- Zhang (CCS 2010): knowing current, allows heuristic guessing of next-password: got 17% in, on average, < 5 online guesses

Benefits at best partial & minor … and little/no concrete evidence.

Q: What alternatives could deliver similar gains at far lower cost?

Thank you.



Q: If house insurers suggest we change all our physical door locks every 90 days, just in case someone has a copy of a key: would we do it?

[No; absent strong evidence, the costs far outweigh expected benefits]