CSCI 331:
Introduction to Computer Security

Lecture 7: Password Cracking, part 2

Instructor: Dan Barowy

Williams

Topics

Paper discussion (Oechslin)

Precomputed Hash Chains

Rainbow Tables

Your to-dos

1. Project part 1 **due Sunday 9/29**.
2. Reading response (Aleph One), **due Tuesday 10/1**.
3. Lab 2 **due Sunday 10/13**.

Keyed encryption functions

Paper discussion (Oechslin)

Precomputed Hash Chains

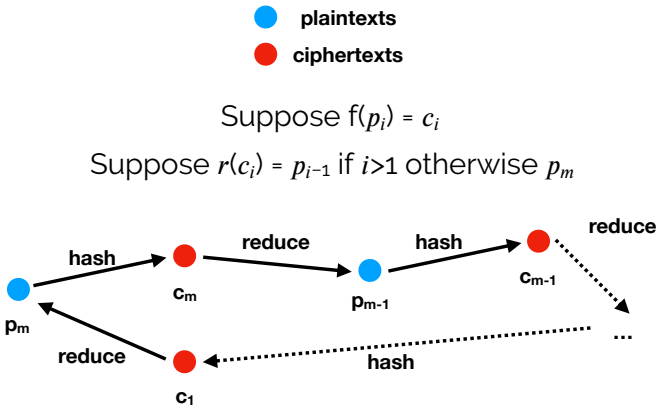
Precomputed Hash Chains

Motivation: dictionaries are **too big** to distribute

Recall:

About 26 terabytes!

Thought experiment

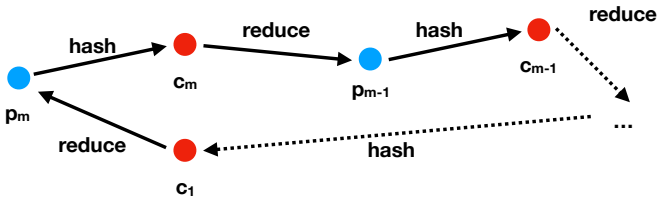


Thought experiment

plaintexts

ciphertexts

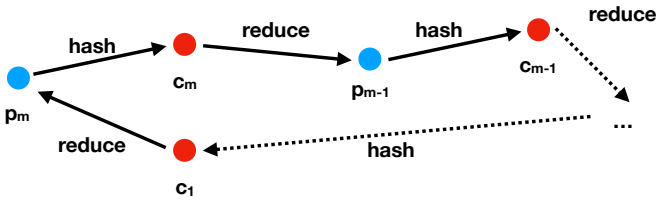
Such a scheme (a *hash chain*) lets us generate all plaintexts (and hashes) from a seed plaintext.



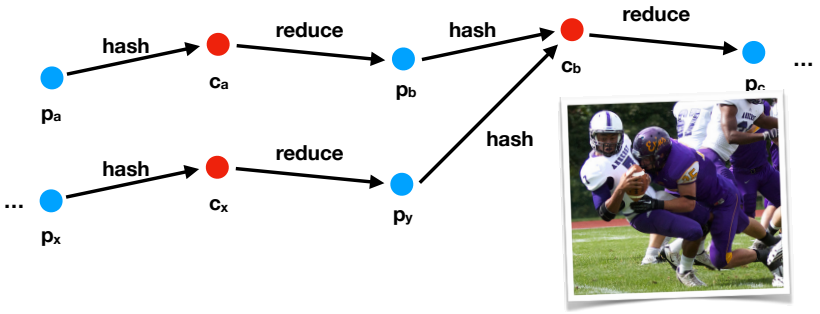
Only need to save the seed. **Drawbacks?**

Thought experiment: **drawbacks**

- Saving just the first password **buys us nothing**. On average, we have to compute **O(m/2)** hash-reductions to find a password.
- It is **probably not possible** to find a reducer that lets you explore the entire password space.
- **Hash functions collide!**



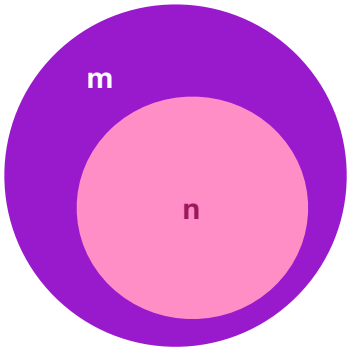
Collisions in a hash chain



After the **collision**, the chain "**loops**."

Collisions prevent us from enumerating the **entire space**!

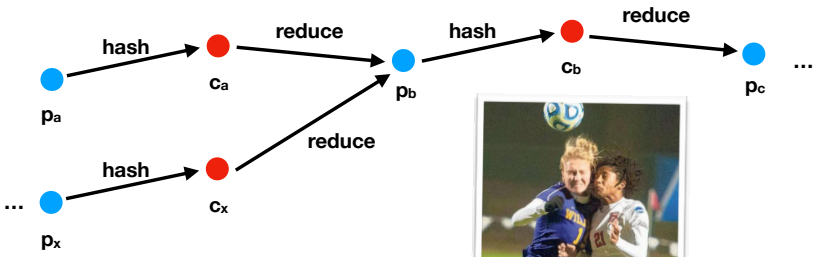
Hashes are guaranteed to collide



m: # of passwords **n**: # of hashes

If **m > n**, we know that **at least (m-n)/m** must collide.

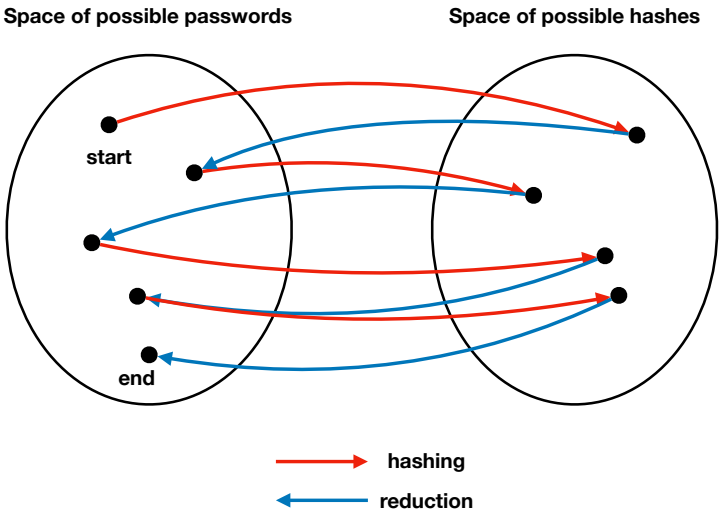
Collisions in a hash chain



Reducers can produce collisions too!

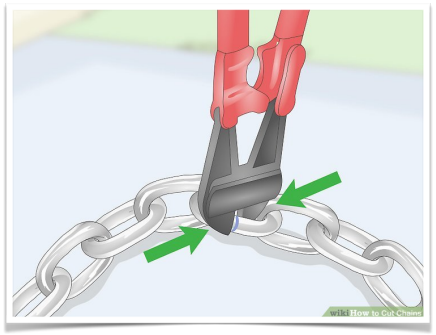
This is what we mean by an **imperfect reducer**.

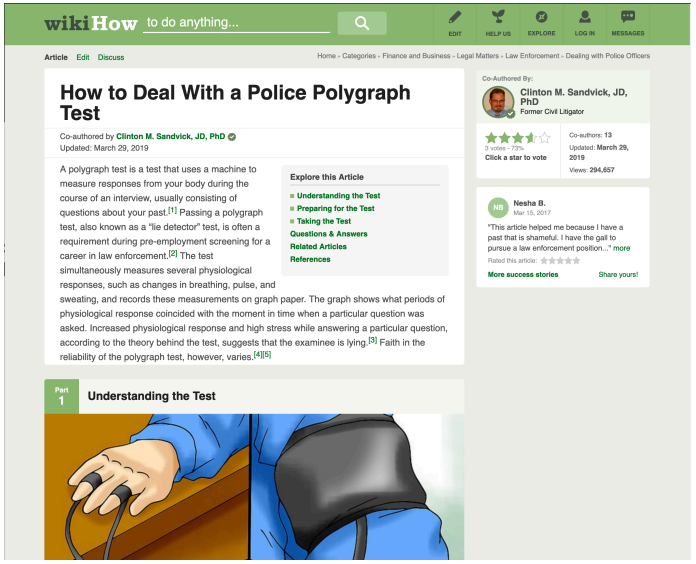
Hash chain



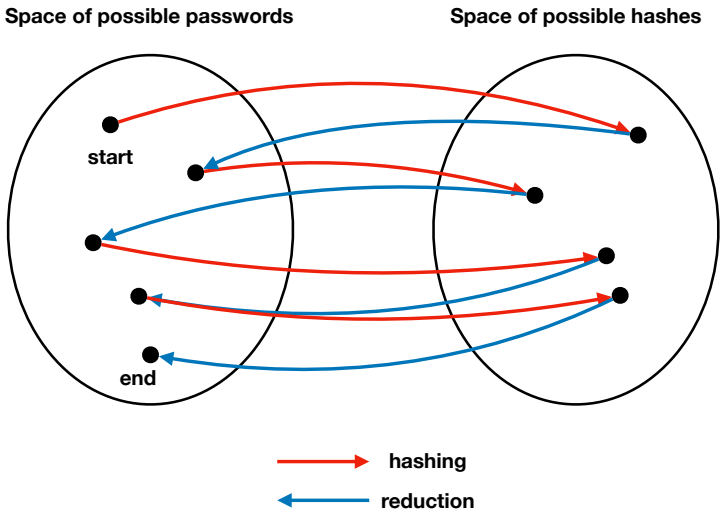
Hash chain of length k

We are going to chop up our long chain into **smaller chains** of length **k**.

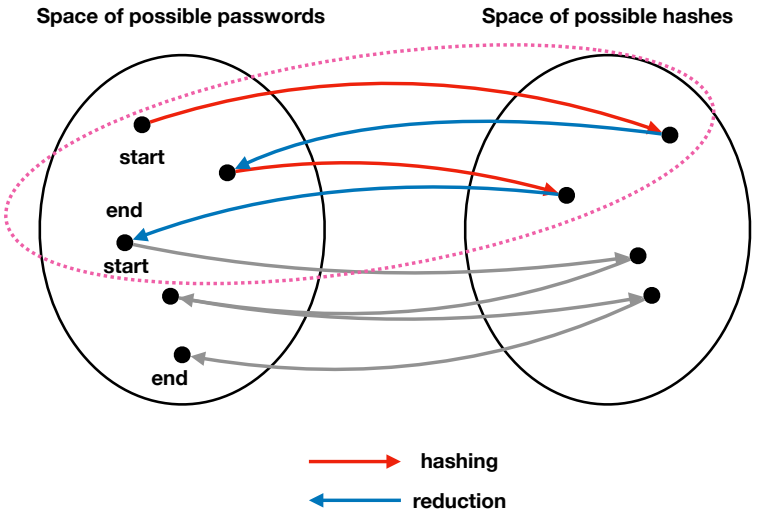




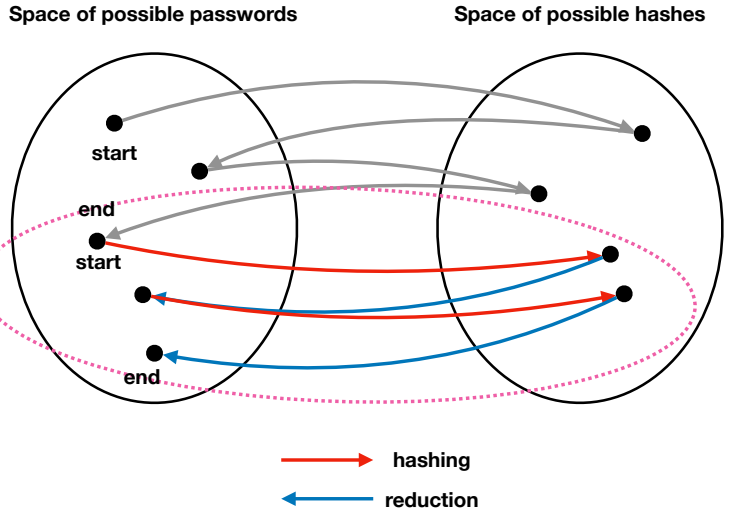
Hash chain



Hash chain of length 2



Hash chain of length 2



Store only start and end

start, end	
p_m	, p_{m-3}
…	
p_5	, p_3
p_3	, p_1

Store it backward

end, start	
p_{m-3}	, p_m
…	
p_3	, p_5
p_1	, p_3

Suppose you are given c_4

end, start
p_{m-3} , p_m
…
p_3 , p_5
p_1 , p_3

c_4 —reduce→ p_3

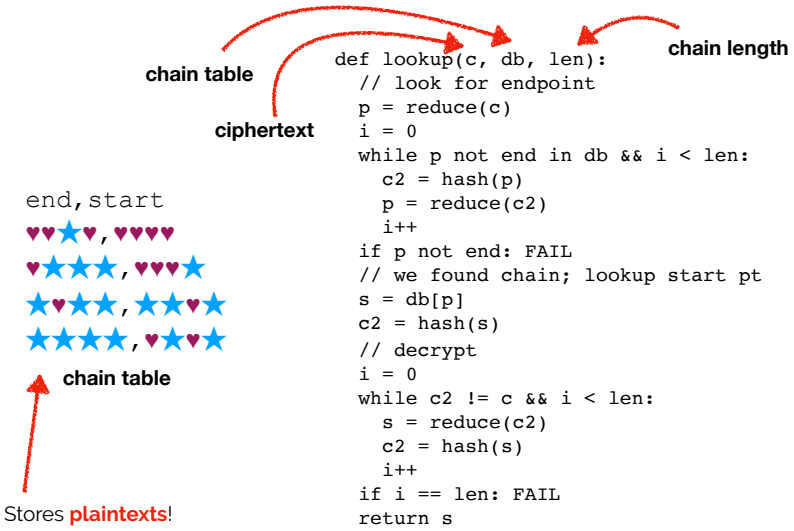
Is p_3 an **end point**? **yes**

Hash and **reduce** from **start point**.

password! original ciphertext

p_5 —**hash**→ c_5 —**reduce**→ p_4 —**hash**→ c_4

Hash chain lookup pseudocode



Class Activity

Decrypt the hash

7F975A56C761DB6506ECA0B37CE6EC87

Answer:

★♥★★

Recap & Next Class

Today we learned:

PCHC algorithm

Next class:

Rainbow algorithm