Implementation of a Rainbow Table Attack

Cybersecurity
Prof. Dr. rer. nat. habil. Clemens H. Cap

Team:

Jasper Roloff
Alexander Saraev
Ovais Idrees
Marvin Davieds
Max Kaseler
Alexandra Plein
Bilal Shah

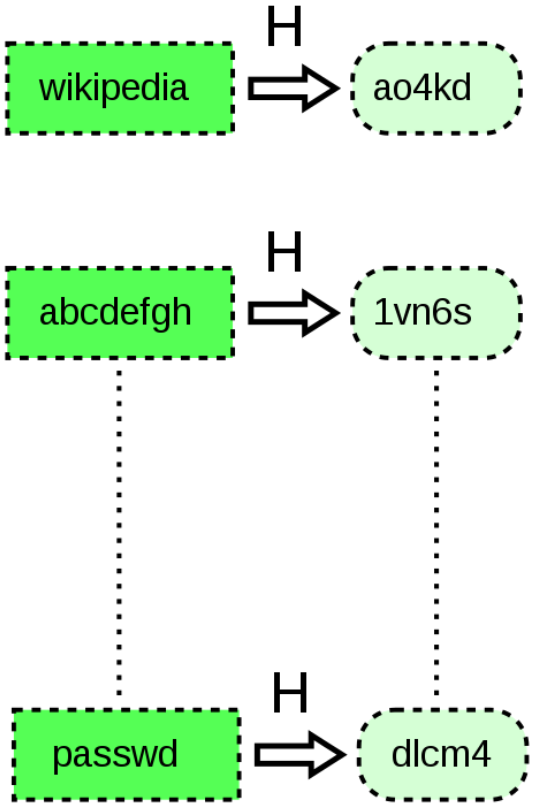
Structure

- Introduction
  - Rainbow Table Attack
  - Task
- Reduction functions
- Implementation
- Task assignment

Rainbow Table Attack

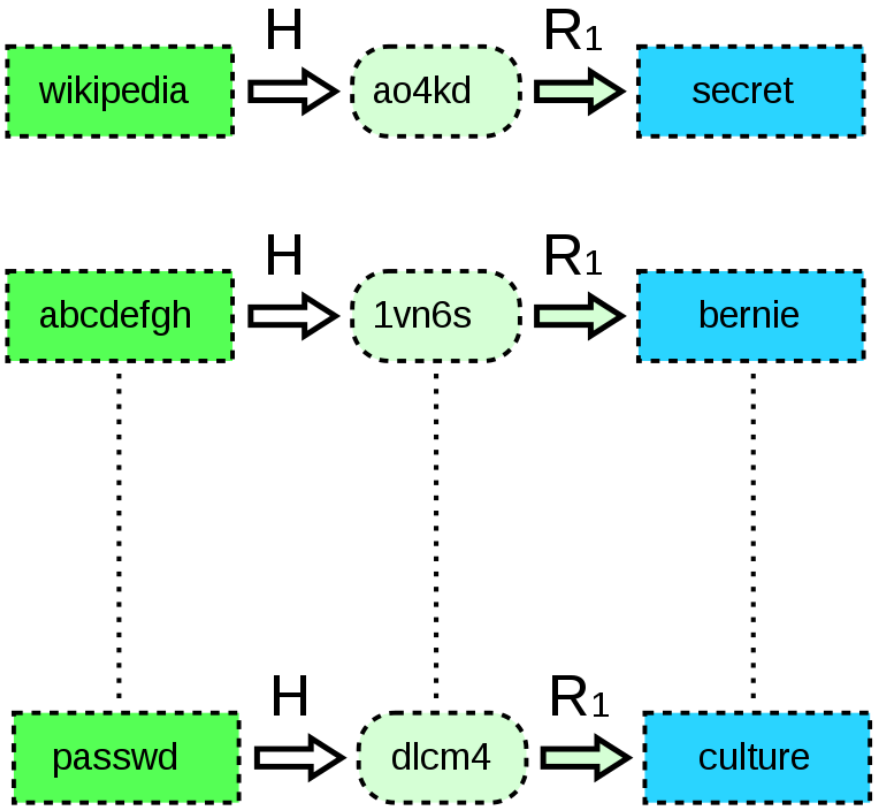
- Precomputed table of password hashes
- Find preimage of a given hash

Rainbow Table



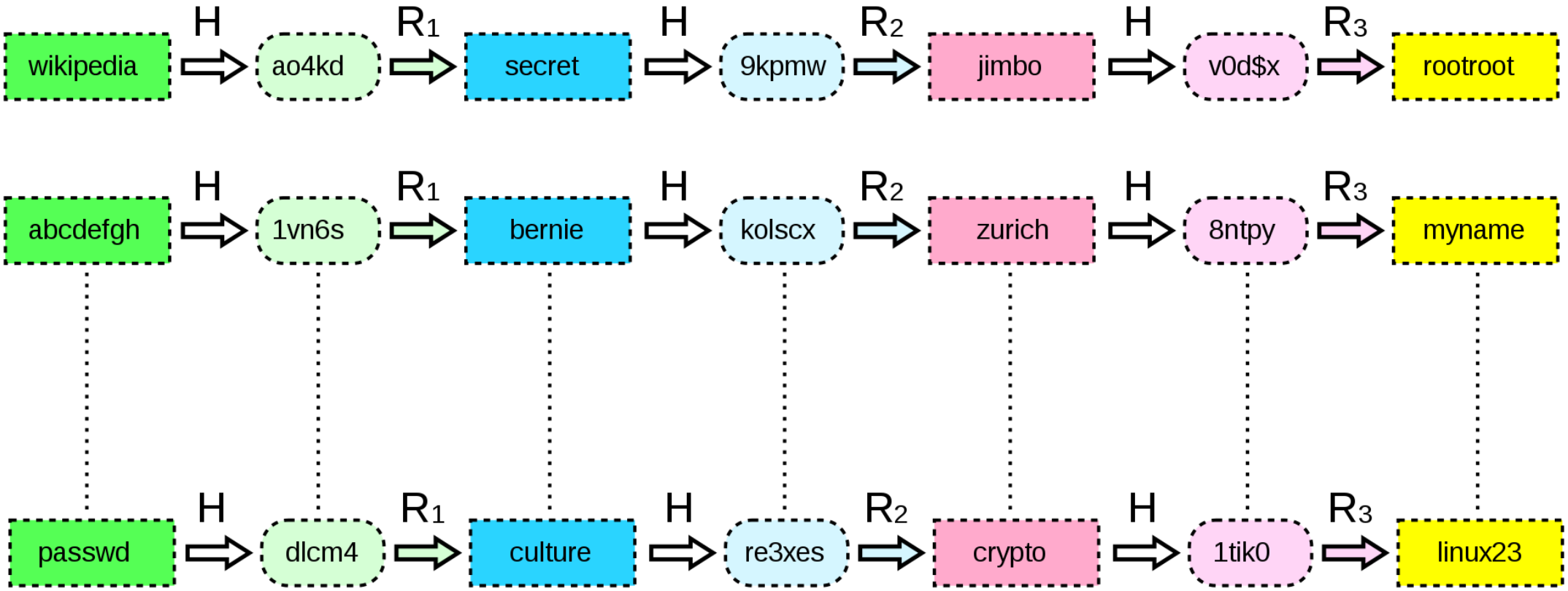
https://de.wikipedia.org/wiki/Rainbow_Table

Rainbow Table



https://de.wikipedia.org/wiki/Rainbow_Table

Rainbow Table



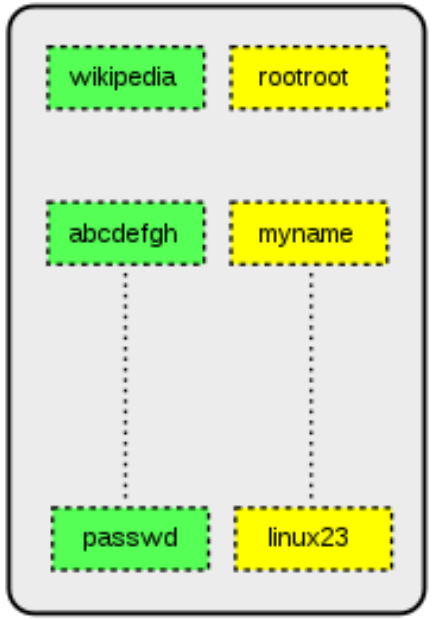
https://de.wikipedia.org/wiki/Rainbow_Table

Rainbow Table

https://de.wikipedia.org/wiki/Rainbow_Table

Rainbow Table Attack

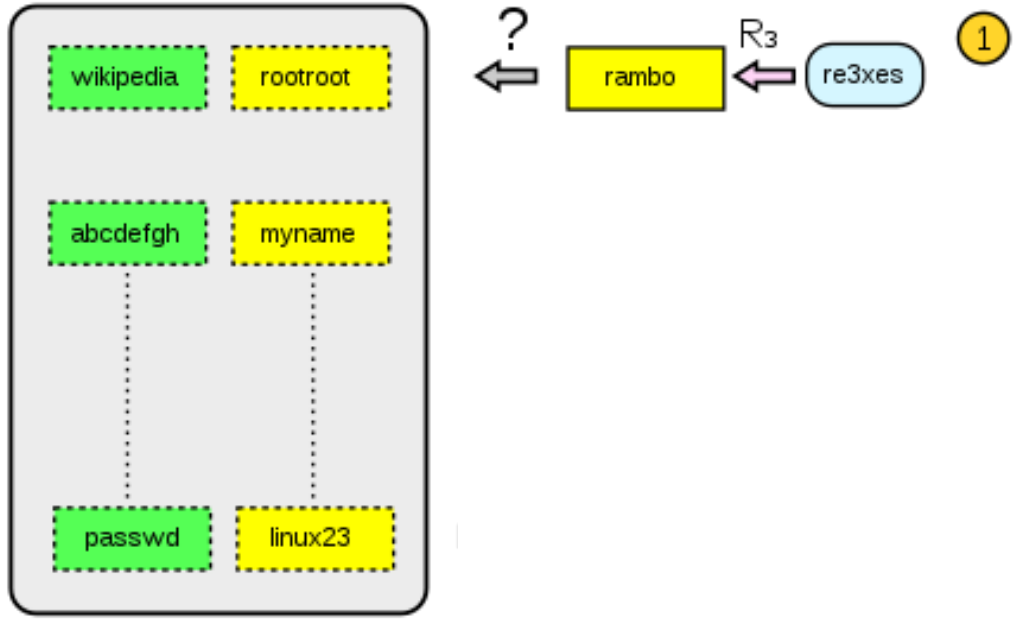
- Goal of attacker: get preimage (plaintext password) of a given hash
- Example: find preimage of hash „re3xes“



https://en.wikipedia.org/wiki/Rainbow_table

Rainbow Table Attack

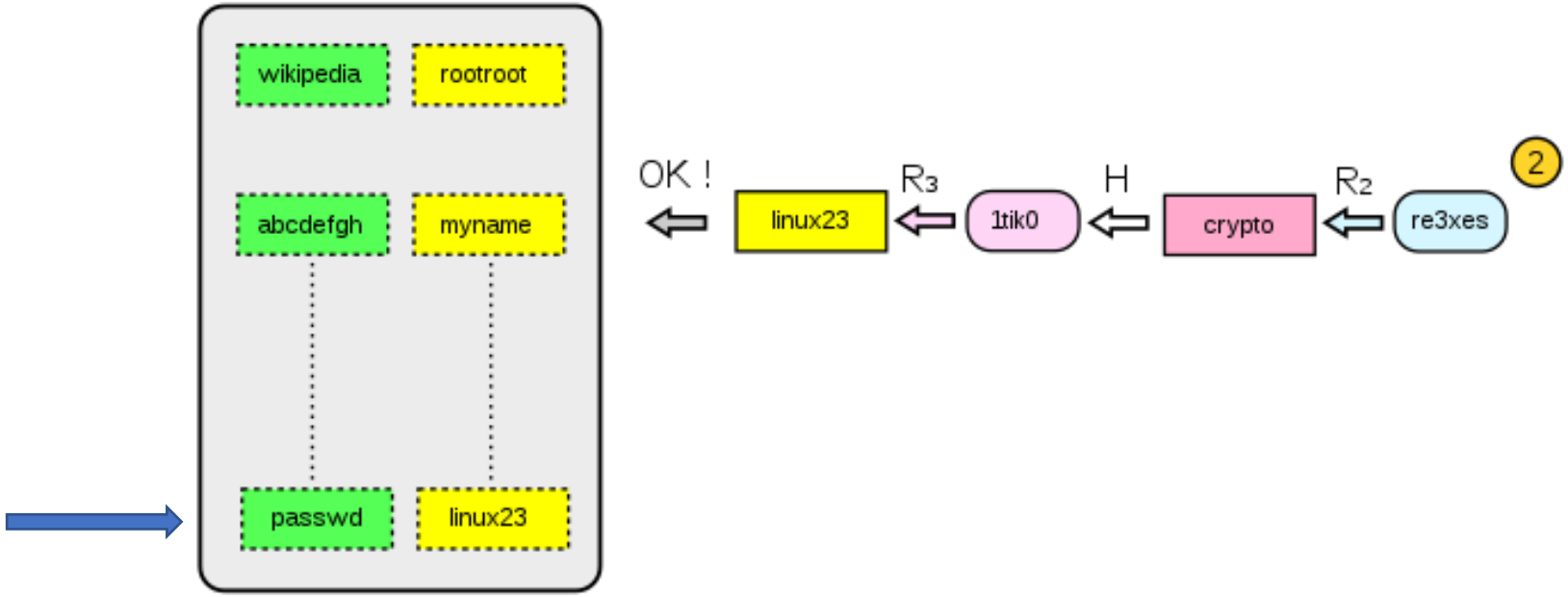
- Goal of attacker: get preimage (plaintext password) of a given hash
- Example: find preimage of hash „re3xes“



https://en.wikipedia.org/wiki/Rainbow_table

Rainbow Table Attack

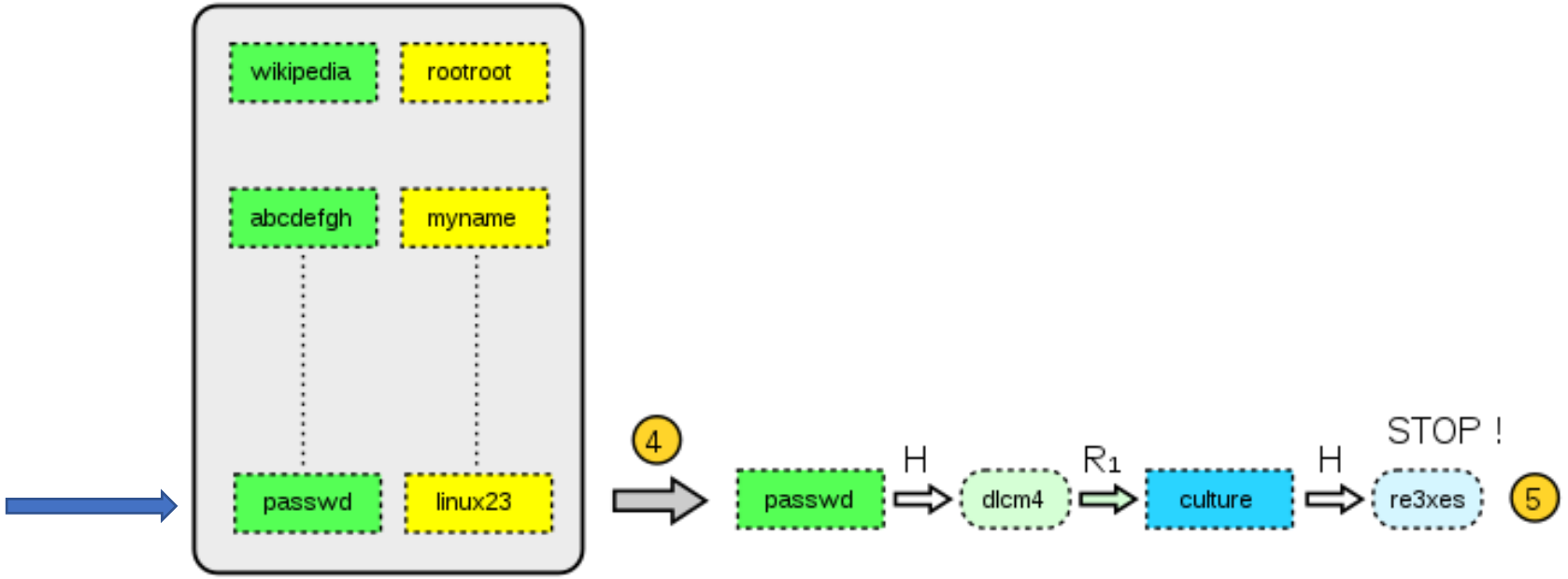
- Goal of attacker: get preimage (plaintext password) of a given hash
- Example: find preimage of hash „re3xes“



https://en.wikipedia.org/wiki/Rainbow_table

Rainbow Table Attack

- Goal of attacker: get preimage (plaintext password) of a given hash
- Example: find preimage of hash „re3xes“



https://en.wikipedia.org/wiki/Rainbow_table

Task

1. Construct a rainbow table (or more than one) using k-prefix weakened SHA-3 (truncated SHA-3)
2. Launch the attack on a given hash-value
3. Adjust k to meet computation time target, 30 – 40 minutes

- Our assumptions:
  - Password length 8
  - lowercase letters only (26 characters)

Reduction Functions

- Hash value → new possible plaintext passwords
- Most common way:
  - f = *toLowercase*(*binary*(hash_value) *mod* searchset_size)
  - different reduction functions: add index of the function to hash value before reducing
- Other way:
  - Take pairs of *num_digits*(hash_value) *mod* alphabet_size
  - Convert every result back into corresponding letters of alphabet
    (0 = a, 1 = b, … , 25 = z)
  - different reduction functions: e.g. take triples instead of pairs or random sample of digits

Implementation

- Python
  - Rainbow-Table data structure
  - Rainbow-Table generation
  - Reduced Hash-Function
  - Automated test for finding proper k
- Pull Requests with peer-review and -approval

Task Assignment