

Implementation of a Rainbow Table Attack

Cybersecurity

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

Team:

Jasper Roloff

Max Kaseler

Alexander Saraev

Alexandra Plein

Ovais Idrees

Bilal Shah

Marvin Davieds

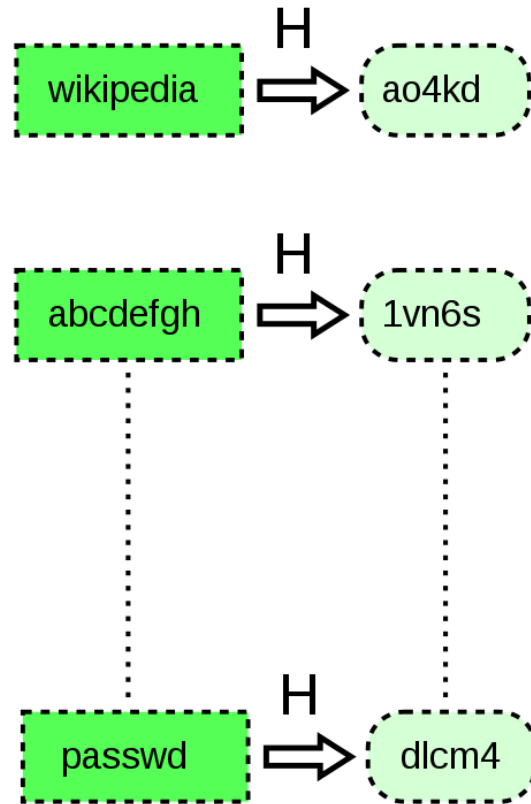
Structure

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

Rainbow Table Attack

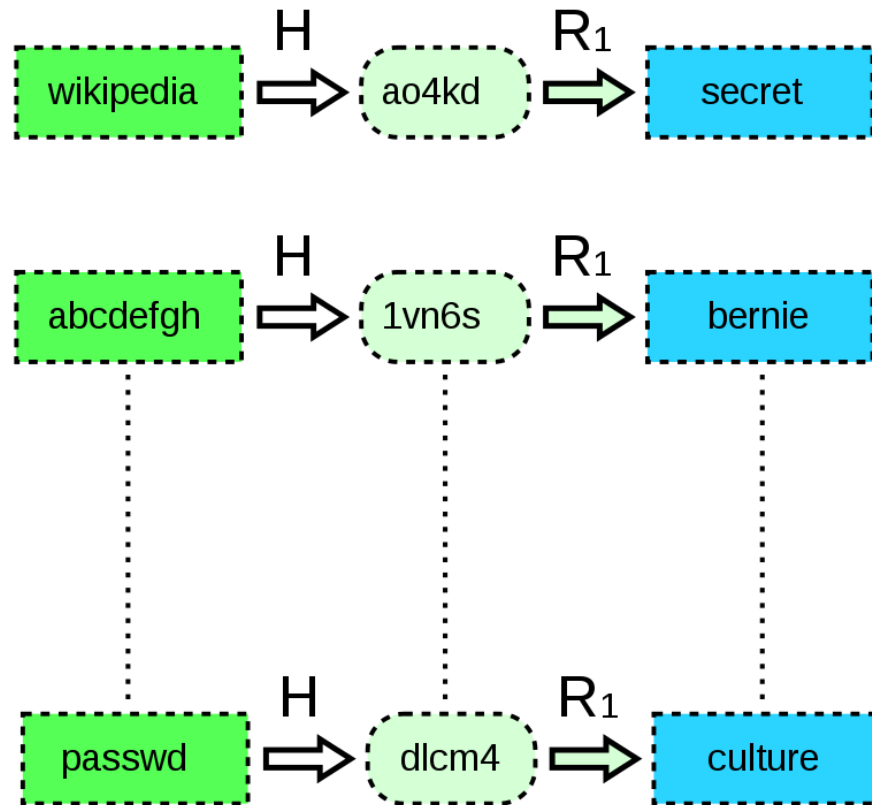
- Precomputed table of password hashes
- Find preimage of a given hash

Rainbow Table



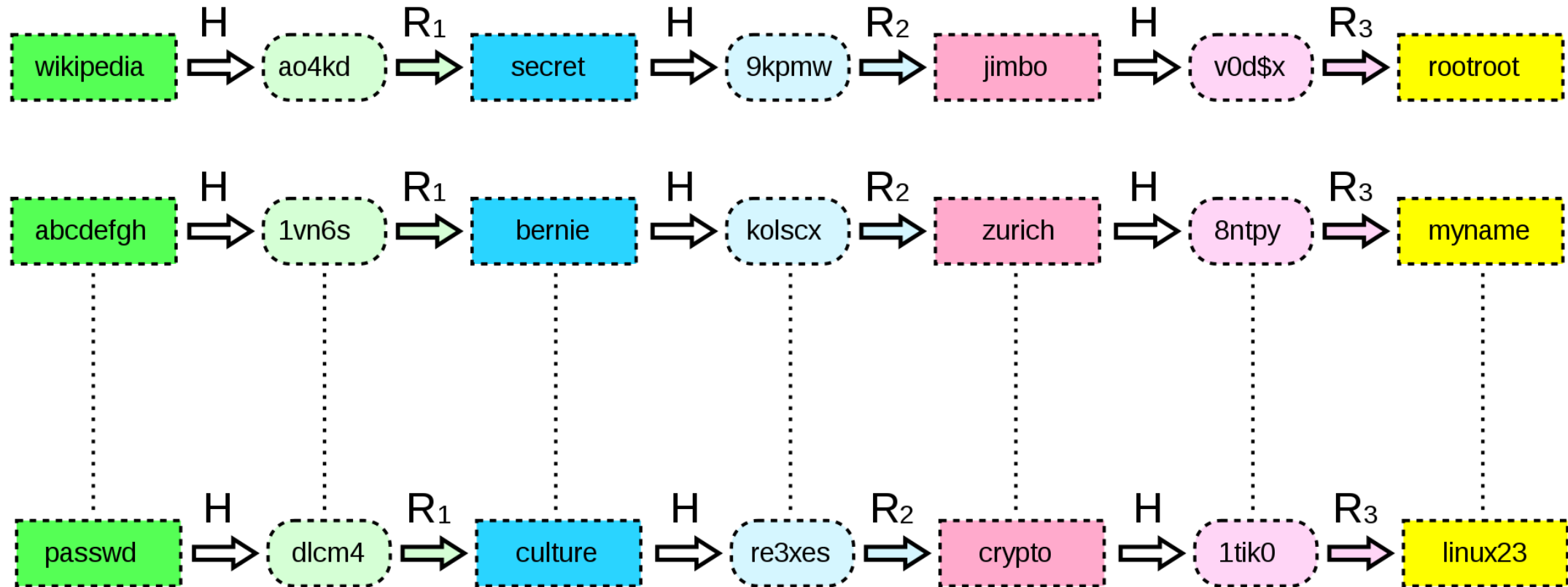
https://de.wikipedia.org/wiki/Rainbow_Table

Rainbow Table



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

Rainbow Table



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

Rainbow Table

wikipedia

abcdefgh

passwd

rootroot

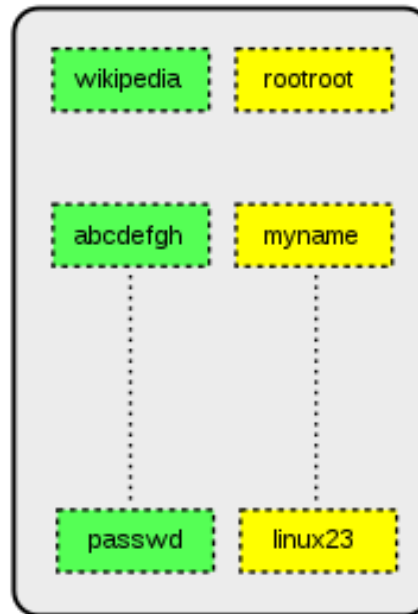
myname

linux23

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

Rainbow Table Attack

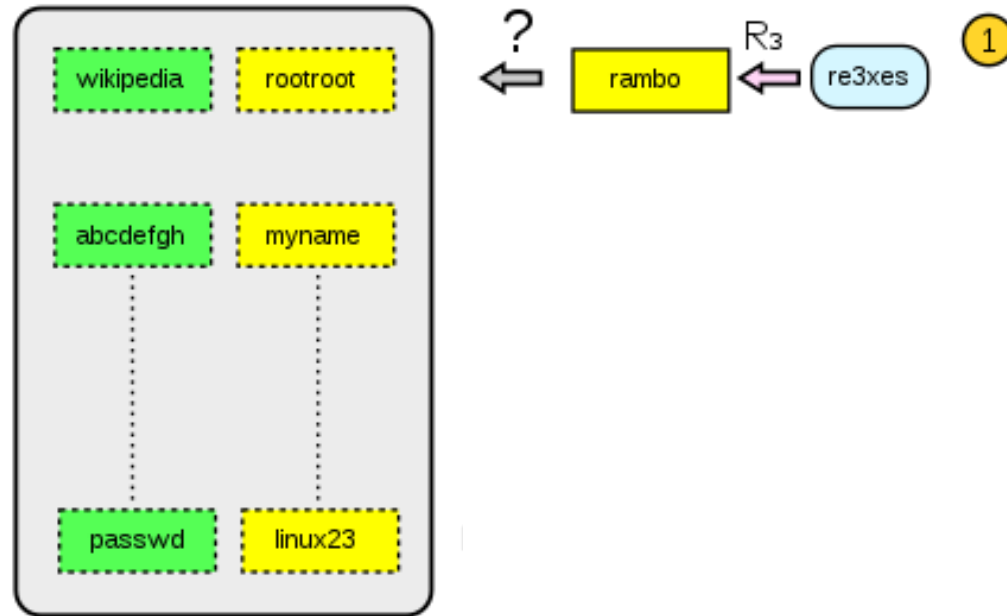
- Goal of attacker: get preimage (plaintext password) of a given hash
- Example: find preimage of hash „re3xes“



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

Rainbow Table Attack

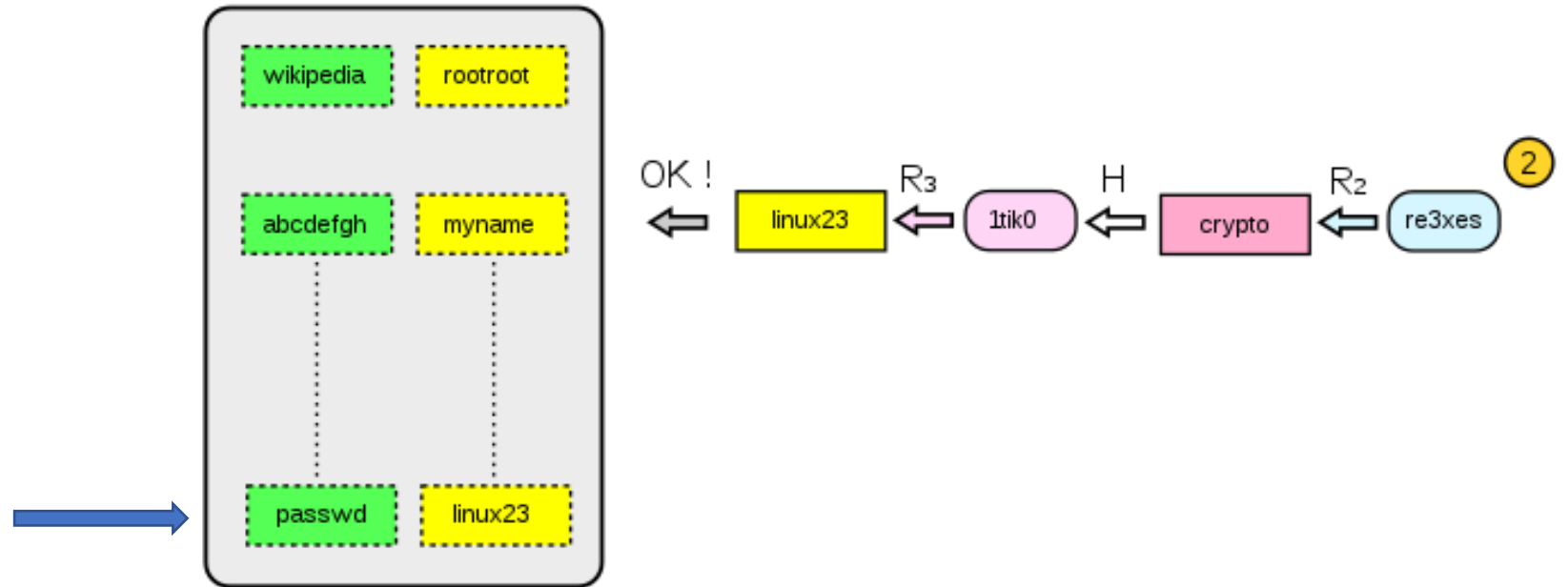
- Goal of attacker: get preimage (plaintext password) of a given hash
- Example: find preimage of hash „re3xes“



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

Rainbow Table Attack

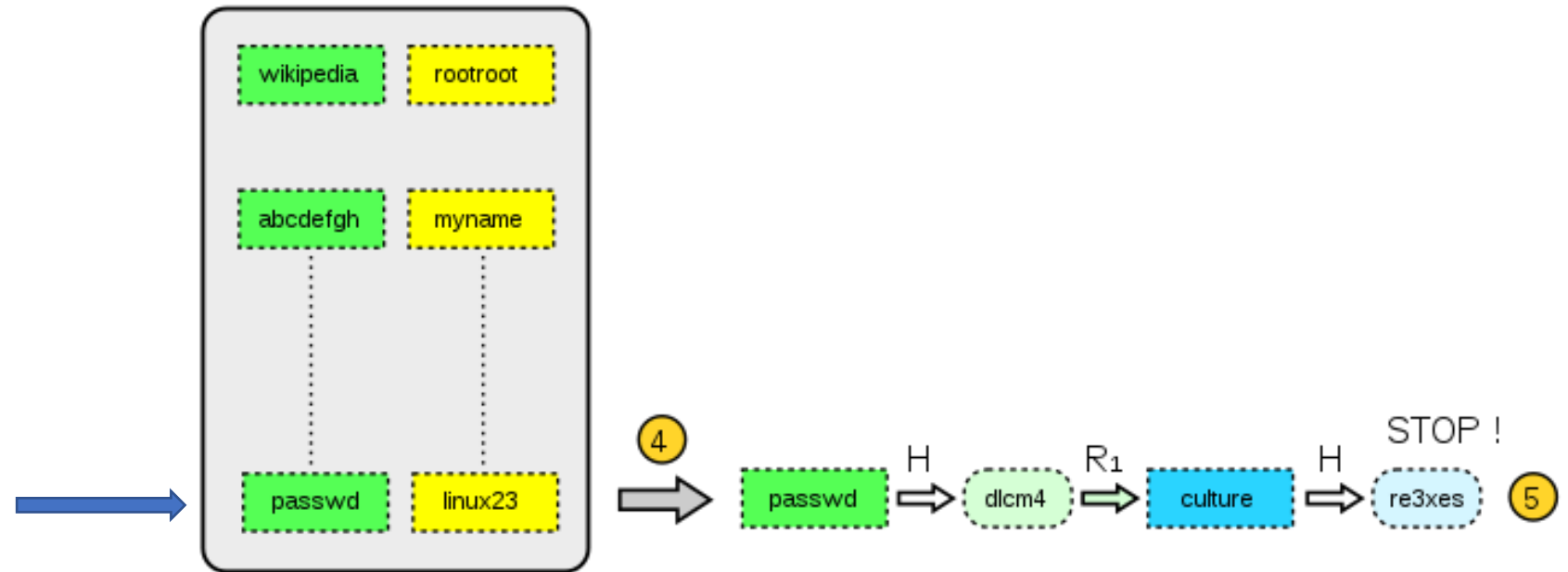
- Goal of attacker: get preimage (plaintext password) of a given hash
- Example: find preimage of hash „re3xes“



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:
 - $f = \text{toLowerCase}(\text{binary}(\text{hash_value}) \bmod \text{searchset_size})$
 - different reduction functions: add index of the function to hash value before reducing
- Other way:
 - Take pairs of $\text{num_digits}(\text{hash_value}) \bmod \text{alphabet_size}$
 - 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

The image shows a screenshot of a task assignment board, likely from a project management tool like Trello. The board is divided into two main columns: 'In Progress' and 'Done'. Each column contains several task cards, each with a title and assigned team members represented by colored avatars. The 'In Progress' column has three cards: 'implementation of the rainbow table itself' (assigned to Marvin and Jasper), 'documentation' (assigned to Ovais and Bilal), and 'automized testing' (assigned to Alexandra). The 'Done' column has two cards: 'presentation slides' (assigned to Marvin, Alexandra, Ovais, Alexander, and Max, with a clock icon and '16. Mai' and a checkmark icon and '4/4') and 'research about concrete reduction functions' (assigned to Alexander and Max). At the bottom of each column is a button to add more cards: '+ Eine weitere Karte hinzufügen'.

Column	Task	Assigned To
In Progress	implementation of the rainbow table itself	Marvin, Jasper
	documentation	Ovais, Bilal
	automized testing	Alexandra
Done	presentation slides	Marvin, Alexandra, Ovais, Alexander, Max
	research about concrete reduction functions	Alexander, Max