Saving Passwords secure

Table of Contents

- Why this presentation?? eHarmony, LinkedIn, last.fm, Sony, ...
- History of saving passwords
  - plaintext
  - hashed passwords
    - broken hash algorithms
    - rainbowtables
    - collisions
  - KDF: key derivation function
- Recommendation for technology and passwords

Me

- Fabian Blechschmidt (@Fabian_ikono)
- blechschmidt@fabian-blechschmidt.de
- PHP since 2004
- Freelancer since 2008
- Magento since 2011
- Certified Magento Developer
- likes playing around, at the moment
  - Magento and Symfony2
  - passwords, hashing, security

Why this presentation

eHarmony, LinkedIn, last.fm, Sony, ...

- LinkedIn, eHarmony (dating site) and last.fm (streaming site) were hacked
  - Goods: 8 million accounts
- „*I left a bunch of stuff running over night, and have about 50% of all the passwords cracked.*“ (LinkedIn: 6,46 Mio Acc., MD5, unsalted)
- http://erratasec.blogspot.de/2012/06/linkedin-vs-password-cracking.html

Assumption
We will be hacked, sooner or later

- Question is not: Are there security holes?
- **it is: When will they be found?**
- And who will find them?

business value

- **TRUST** is your business value
- If the user doesn‘t trust you, you don‘t make a deal.

the second big problem (for the user)

the second big problem (for the user)

- User uses his password on 100 service

the second big problem (for the user)

- User uses his password on 100 service
- your service is not important and it doesn‘t matter wether the passwords are stolen?

the second big problem (for the user)

- User uses his password on 100 service
- your service is not important and it doesn‘t matter wether the passwords are stolen?
- **REALLY!??!!**

the second big problem (for the user)

- User uses his password on 100 service
- your service is not important and it doesn‘t matter wether the passwords are stolen?
- **REALLY!??!!**
- **He uses THE SAME PASSWORD ON 100 OTHER SERVICES!**

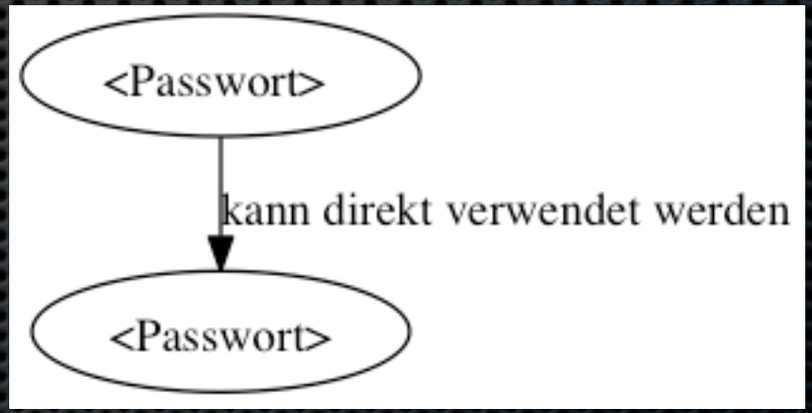
plain text passwords

plain text passwords

- How? Save password directly into database
- Advantage:
  - The user gets his password back
- Disadvantage:
  - database stolen →all passwords stolen

plain text passwords - problems

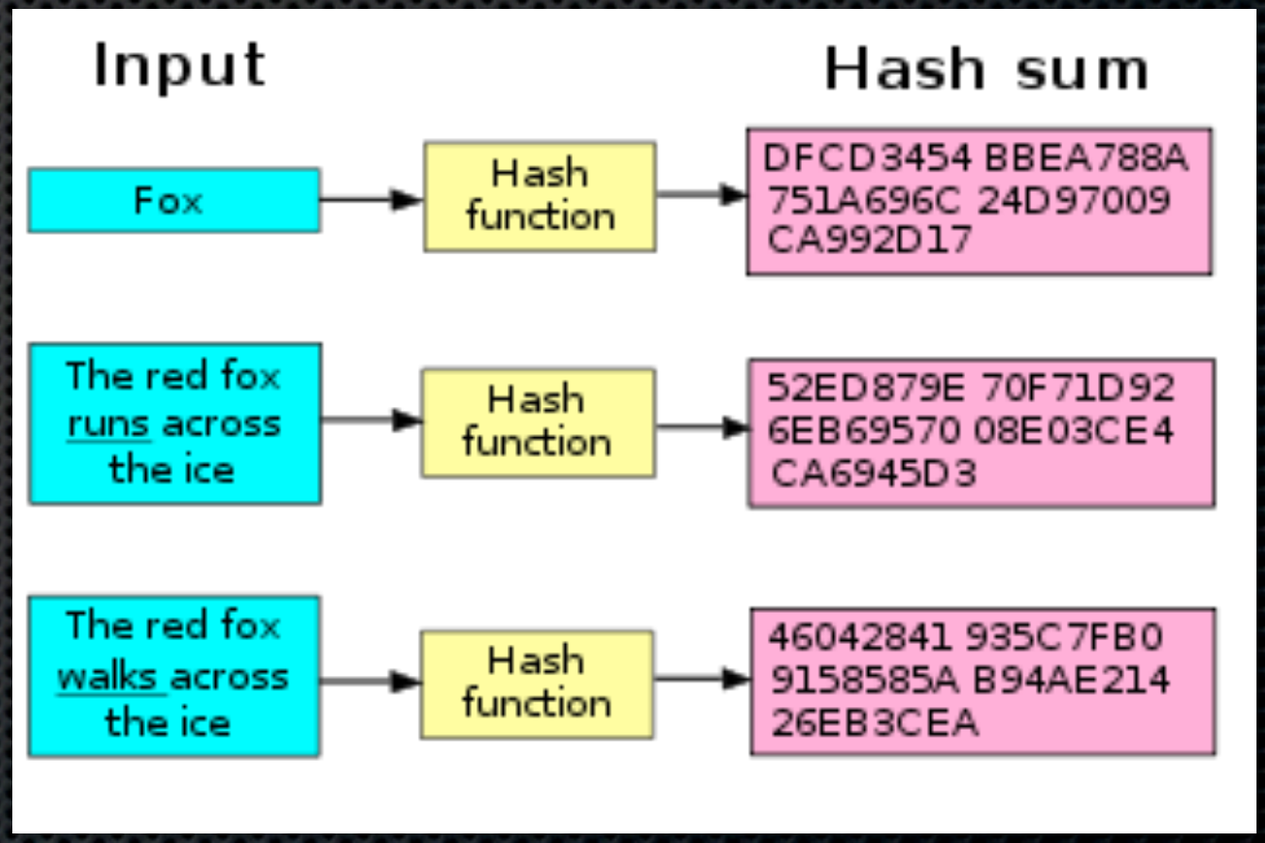
- problems:
  - time for cracking:
    - none.



hashed passwords

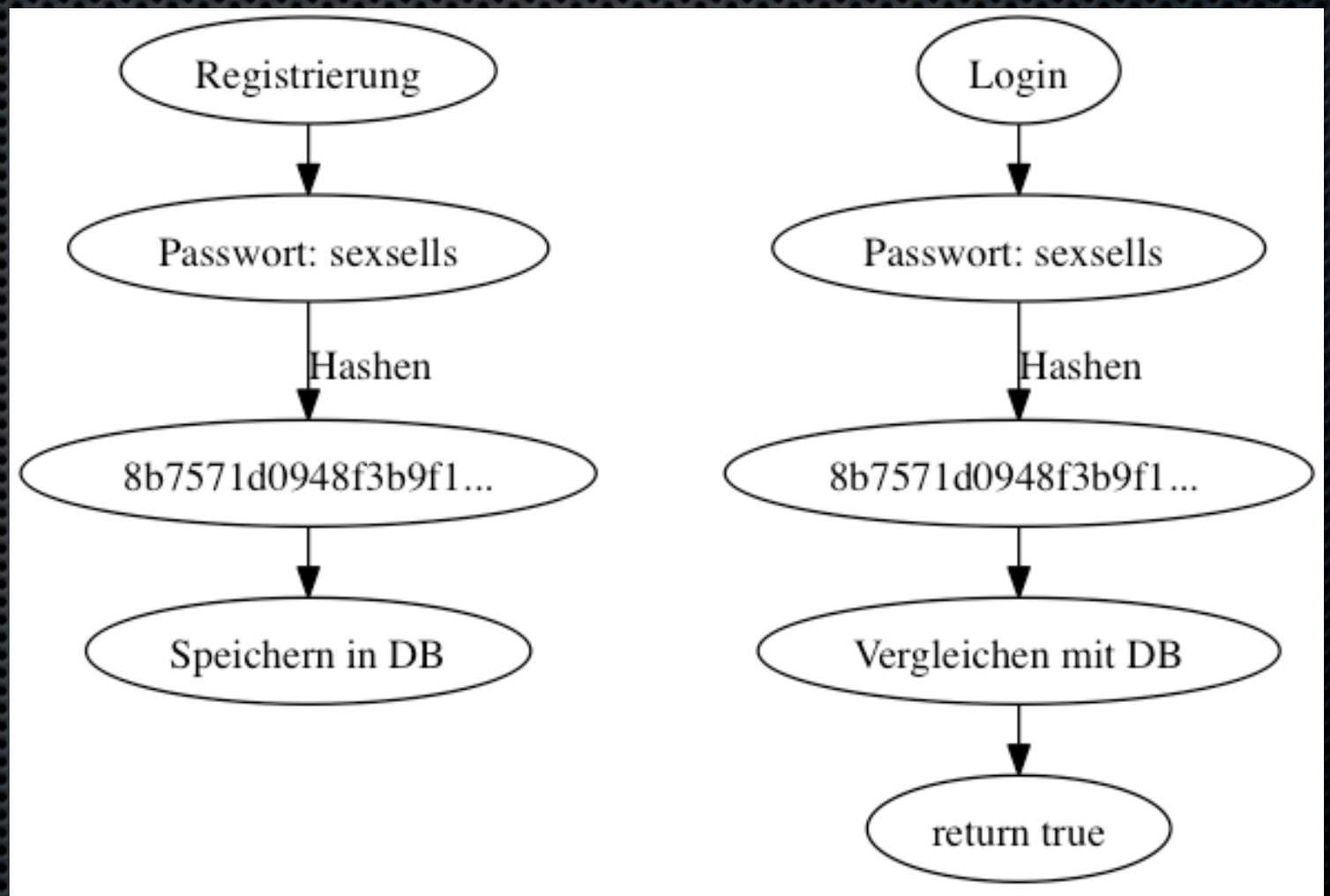
Hashing

- One-Way-Encryption
- easy in one direction
  hard in the other
  - e.g. prime factorization, multiplication vs. modulo
- e.g. SHA2, SHA512, Whirlpool



hashed passwords

- Save hashed passwords
- Advantage:
  - password is not plaintext
- Disadvantage:
  - today still easy to crack



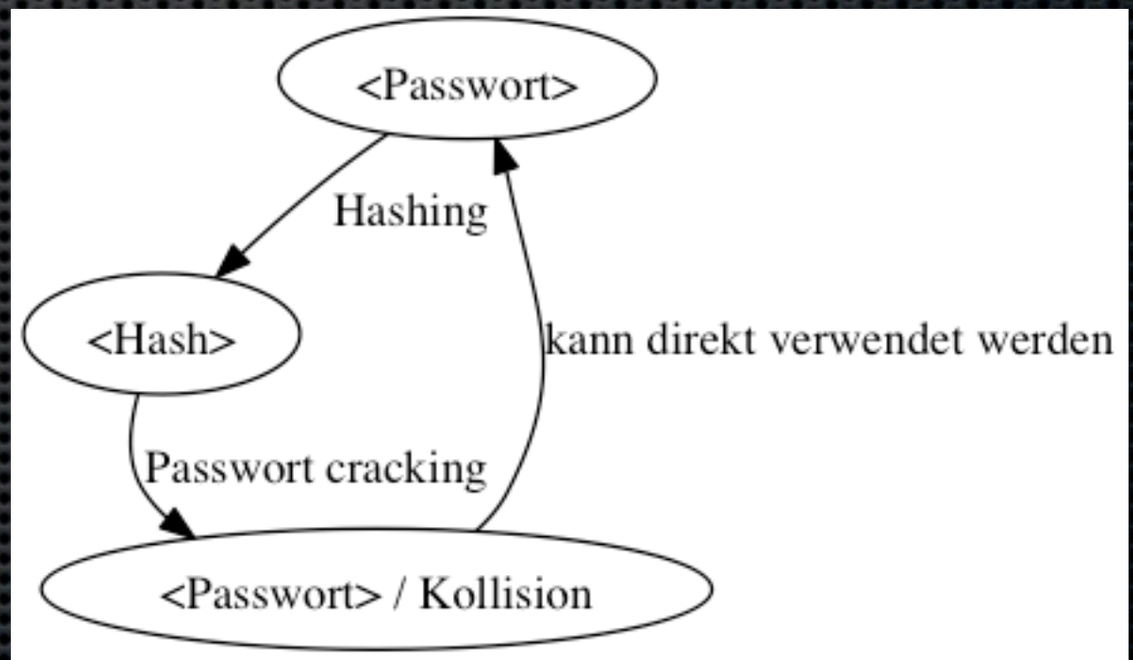
hashed passwords - problems

- finding plaintext password for hash isn‘t easy, but

  (1) many hash-algorithms are broken (DES, MD5, SHA1, ...)

  (2) it is possible to precalculate passwords/collisions (rainbowtables)

  (3) today we have much processing ressources (GPU, EC2, ...)



broken hash algorithms

broken algorithms

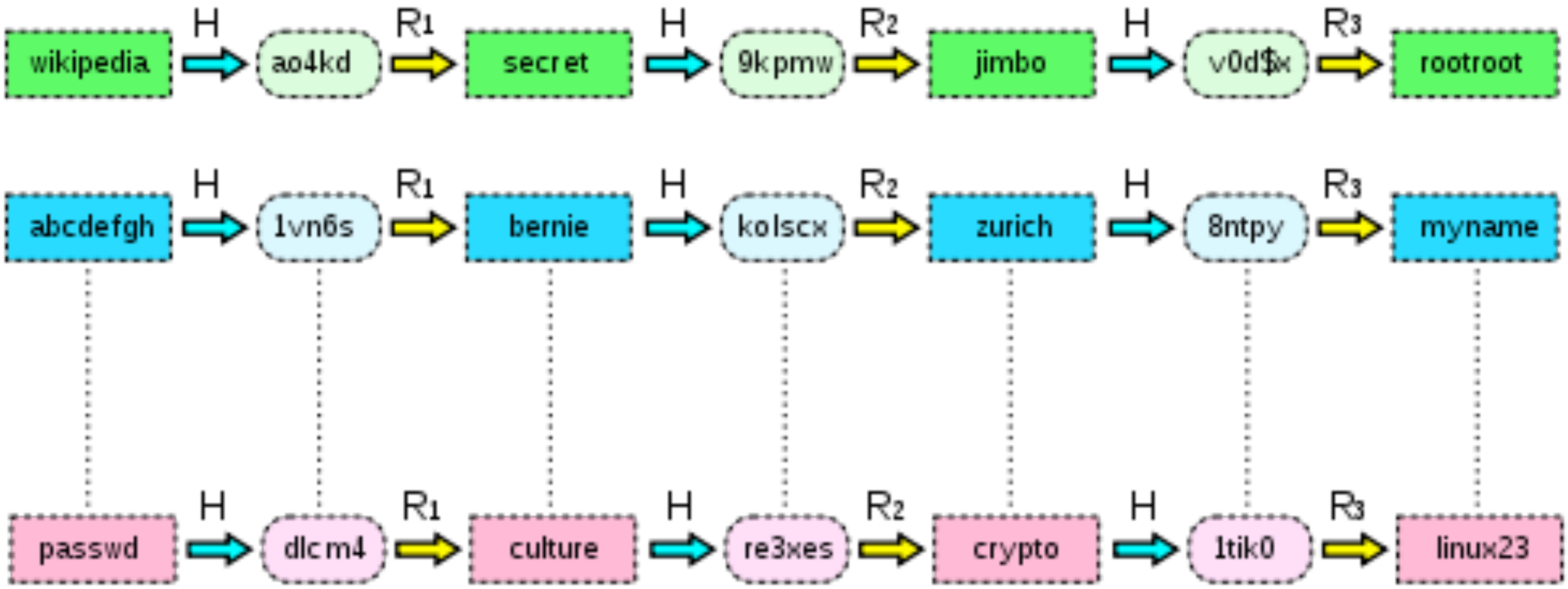
- Hands off!

- DES (old crypt() Impl.)
- MD2
- MD4
- MD5
- SHA1
- RC4 (WEP)

rainbowtables

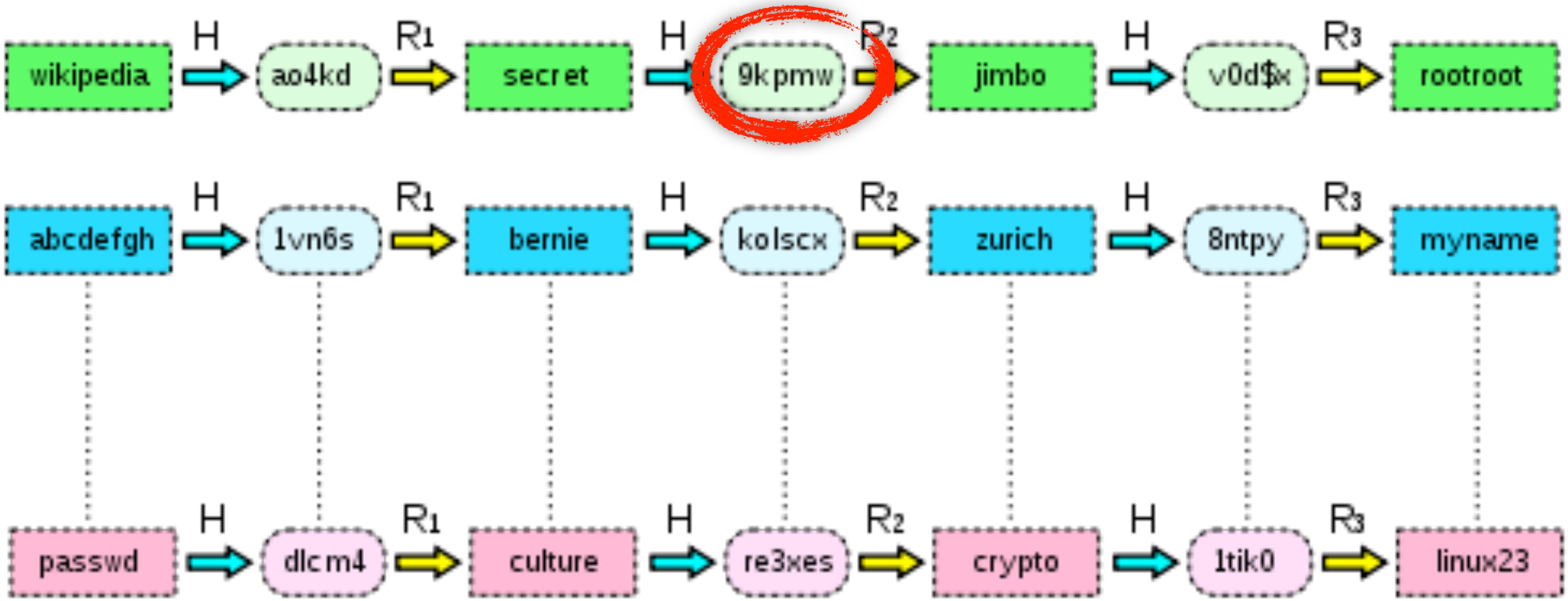
rainbowtables - benefit

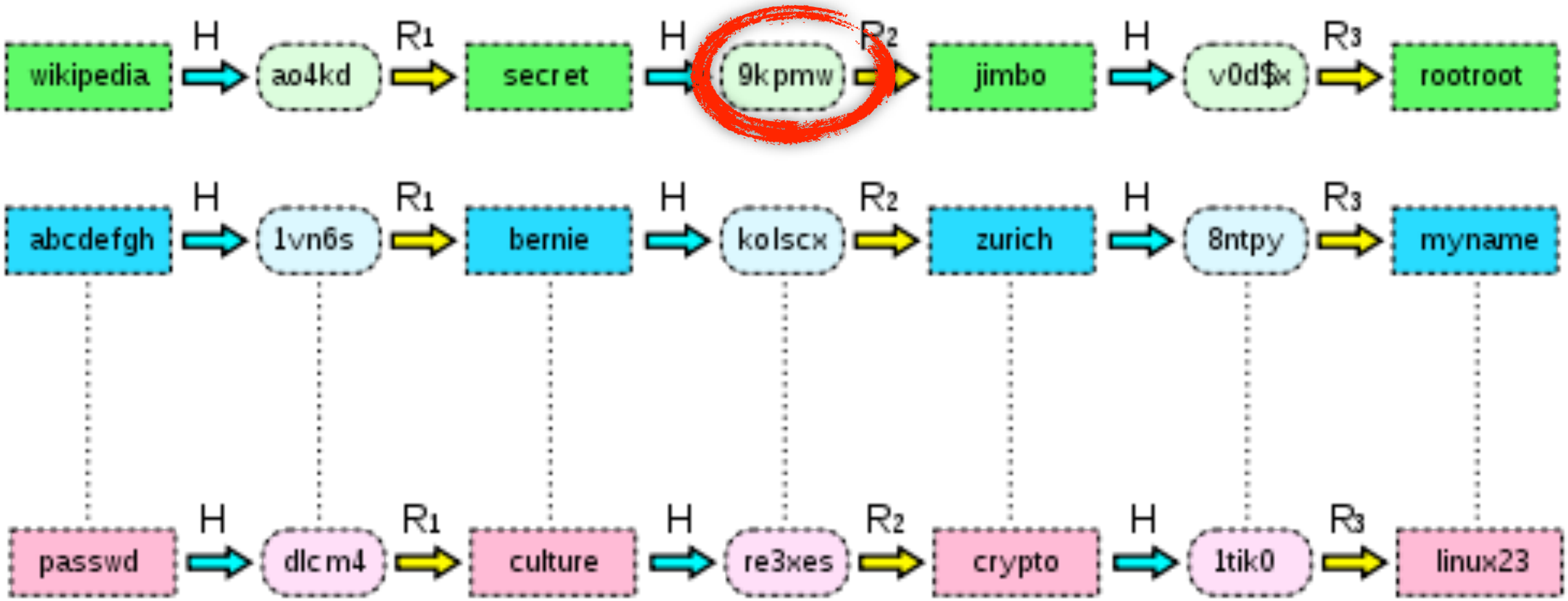
- only first value, reductionfunction and last value is saved
- reduced storage space needed



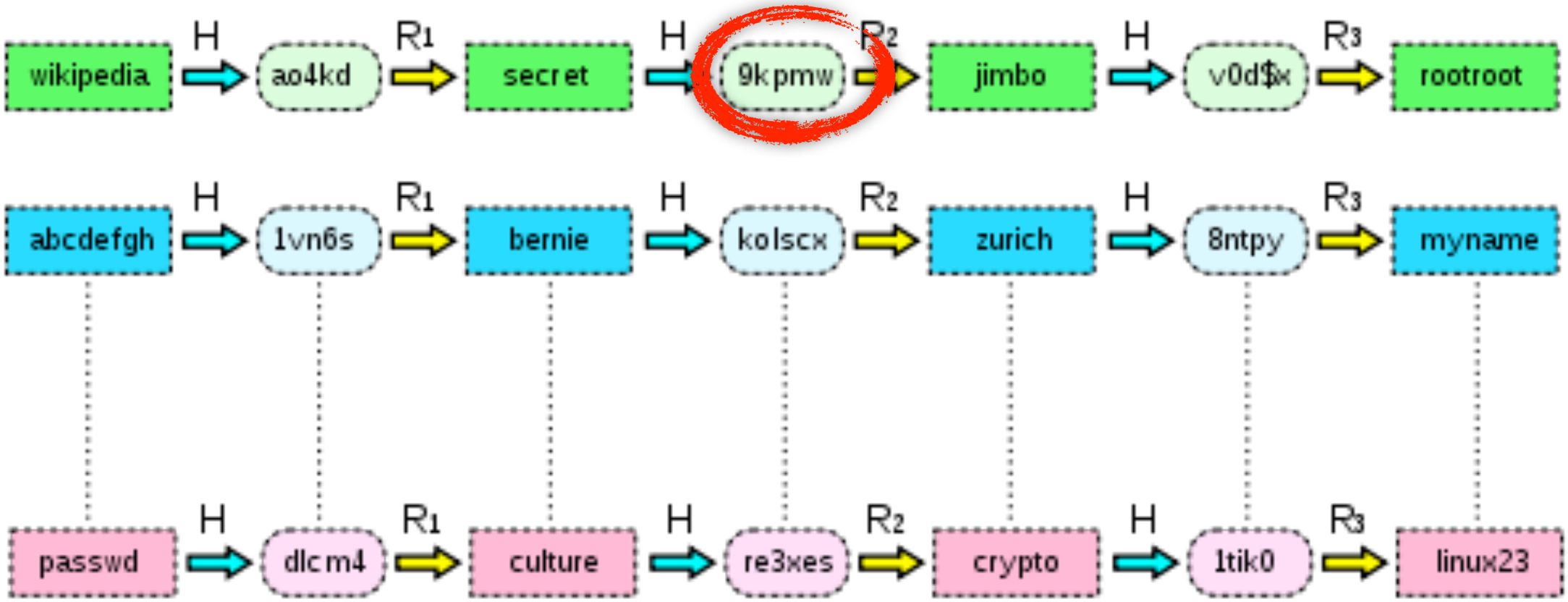
rainbowtable

first value → hashing → hash → reduction → etc.


wikipedia
H
ao4kd
R1
secret
H
9kpmw
R2
jimbo
H
v0d$x
R3
rootroot
abcdefgh
H
1vn6s
R1
bernie
H
kolscx
R2
zurich
H
8ntpy
R3
myname
passwd
H
dlcm4
R1
culture
H
re3xes
R2
crypto
H
1tik0
R3
linux23

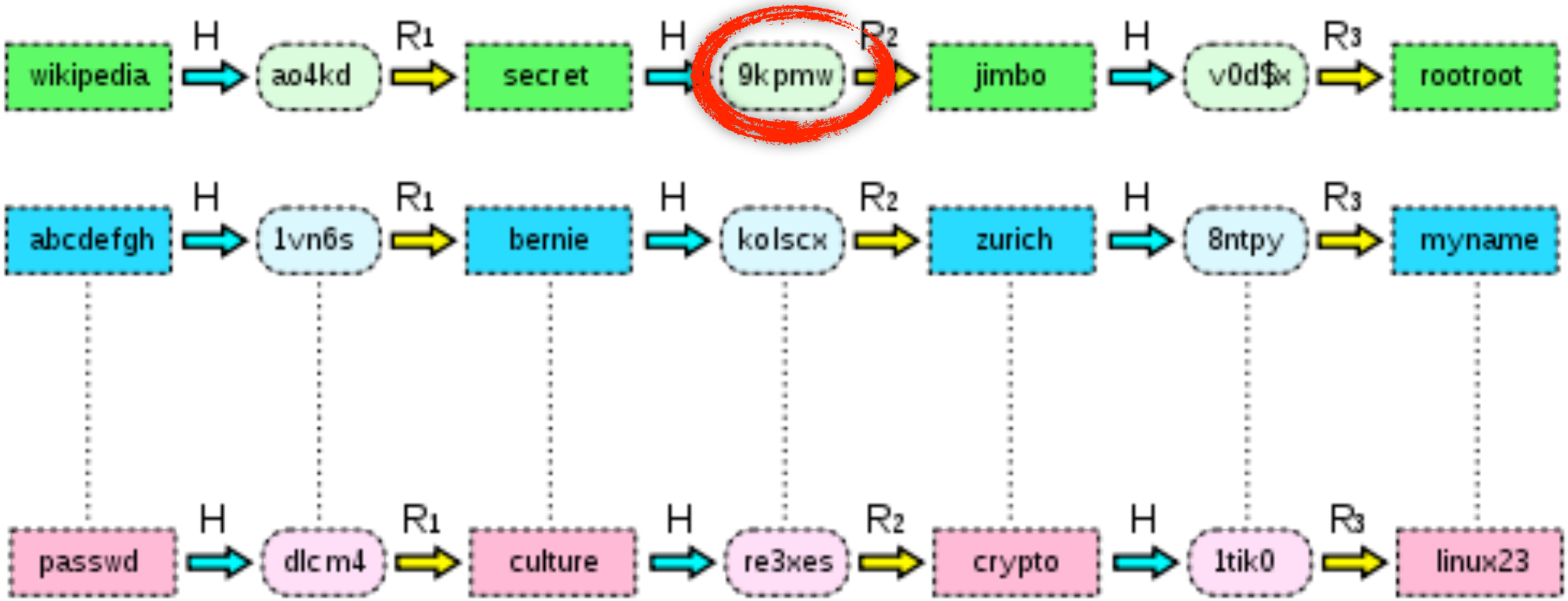


wanted hash: 9kpmw



wanted hash: 9kpmw

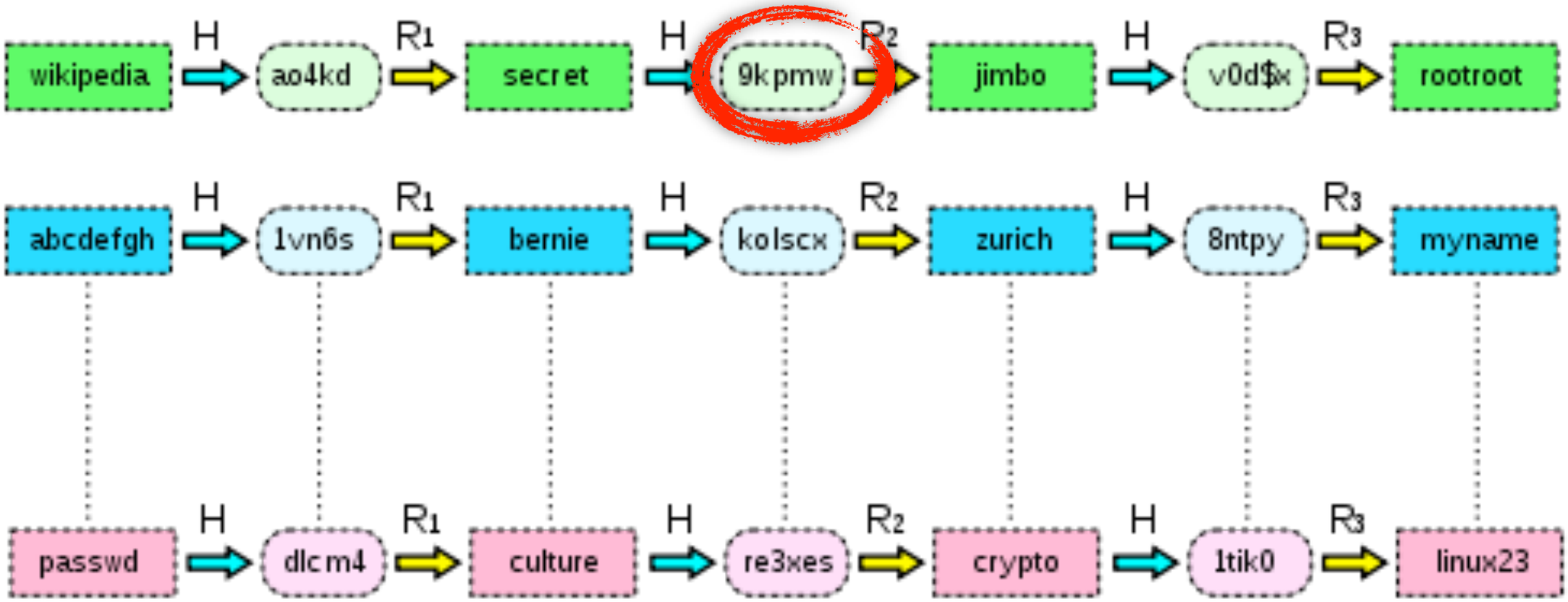
9kpmw →R2→jimbo→H→v0d$x→R3→rootroot



wanted hash: 9kpmw

9kpmw →R2→jimbo→H→v0d$x→R3→rootroot

rootroot→START→wikipedia

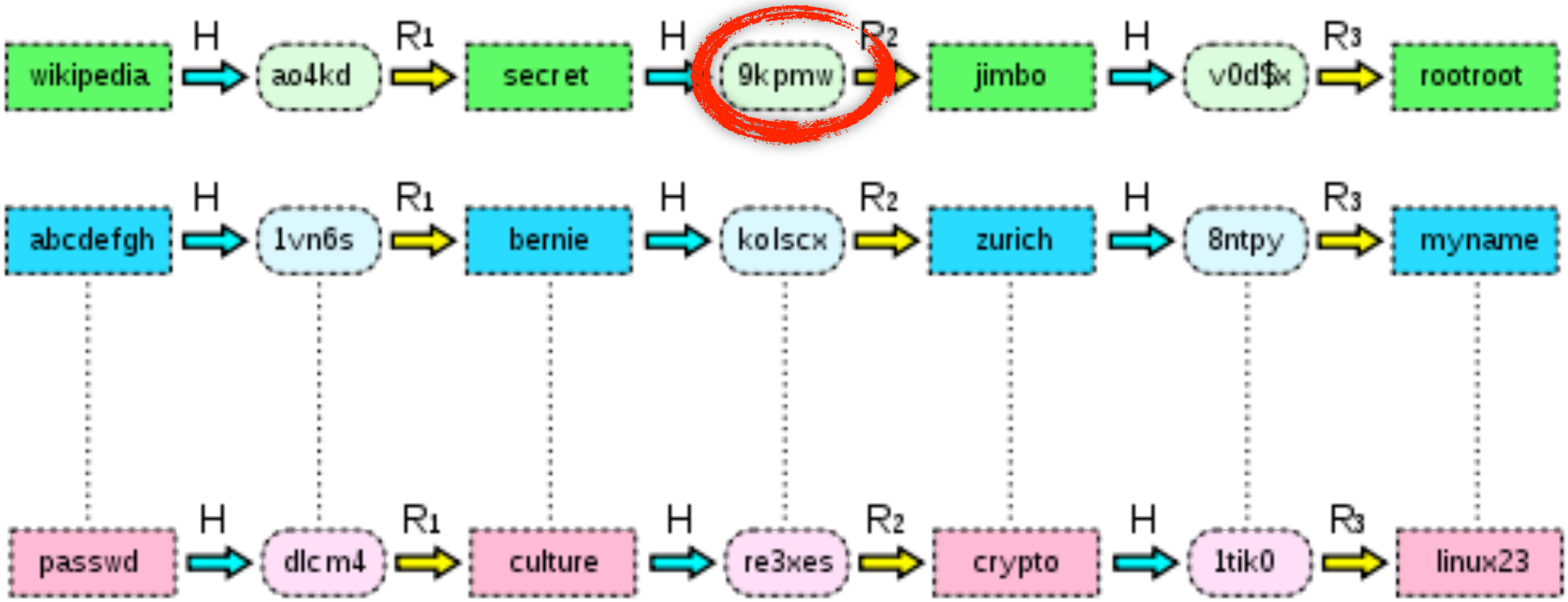


wanted hash: 9kpmw

9kpmw →R2→jimbo→H→v0d$x→R3→rootroot

rootroot→START→wikipedia

wikipedia→H→ao4kd→R1→secret→9kpmw



wanted hash: 9kpmw

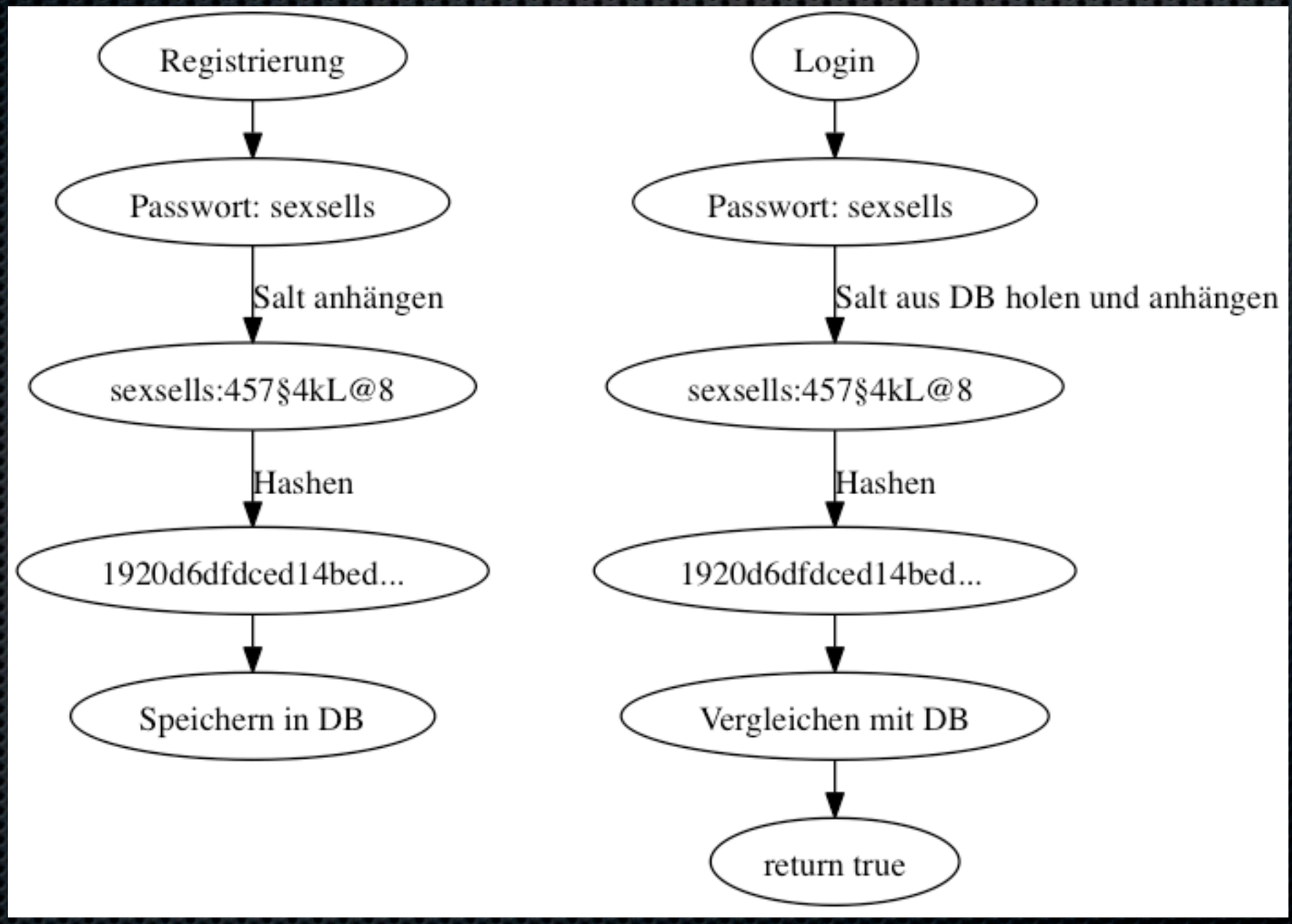
9kpmw →R2→jimbo→H→v0d$x→R3→rootroot

rootroot→START→wikipedia

wikipedia→H→ao4kd→R1→secret→9kpmw

FOUND: **9kpmw**

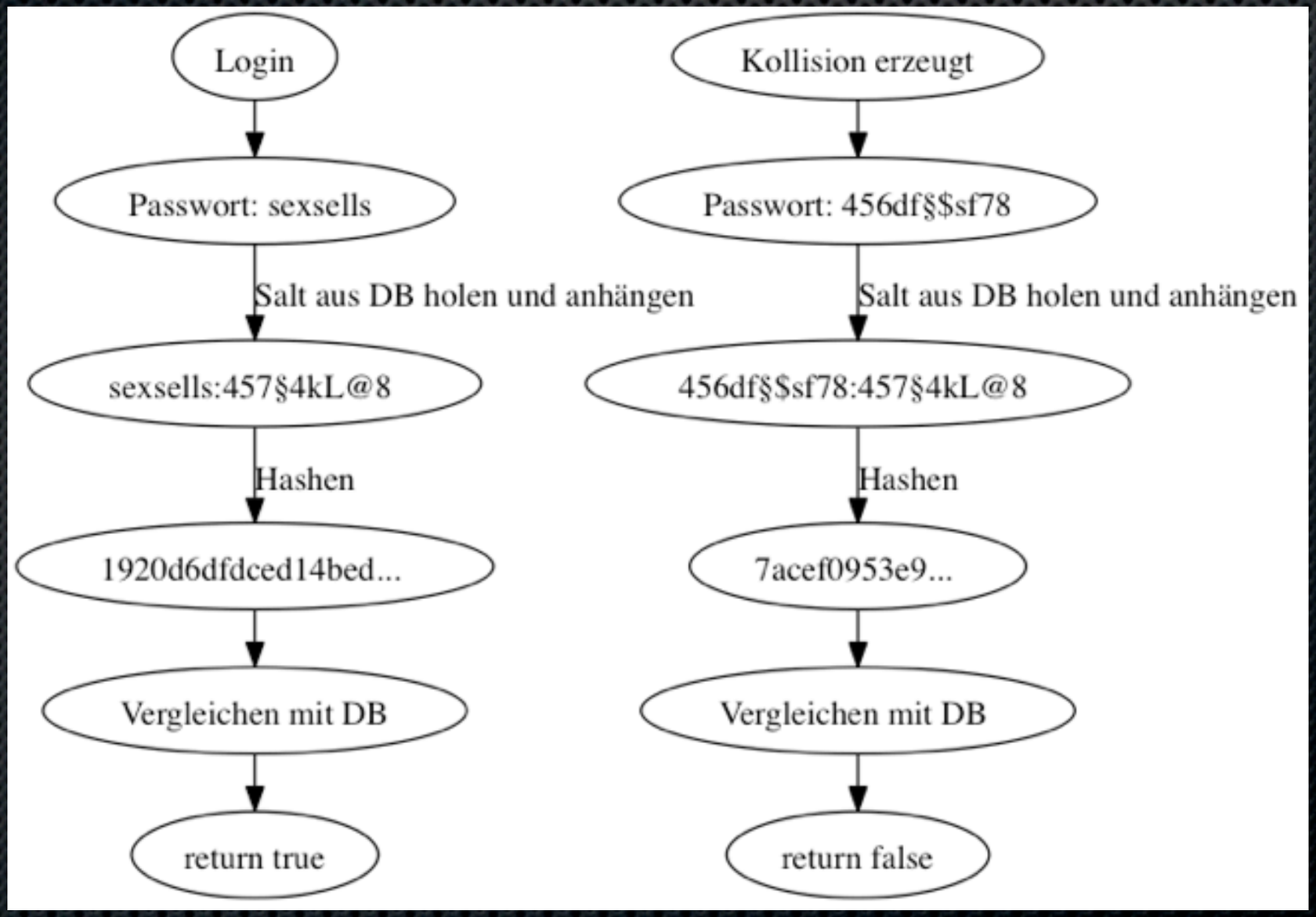
problem: rainbowtables and collision
solution: salting



works against rainbowtables

- big rainbowtables only exist for unsalted hashes
- every salt needs his own rainbowtable
  - because prefix/suffix is static
- table sizes expands (very very much)
  - goal is not to avoid, it‘s to raise costs and time

"H(sexsells) == H(456df§$sf78)"

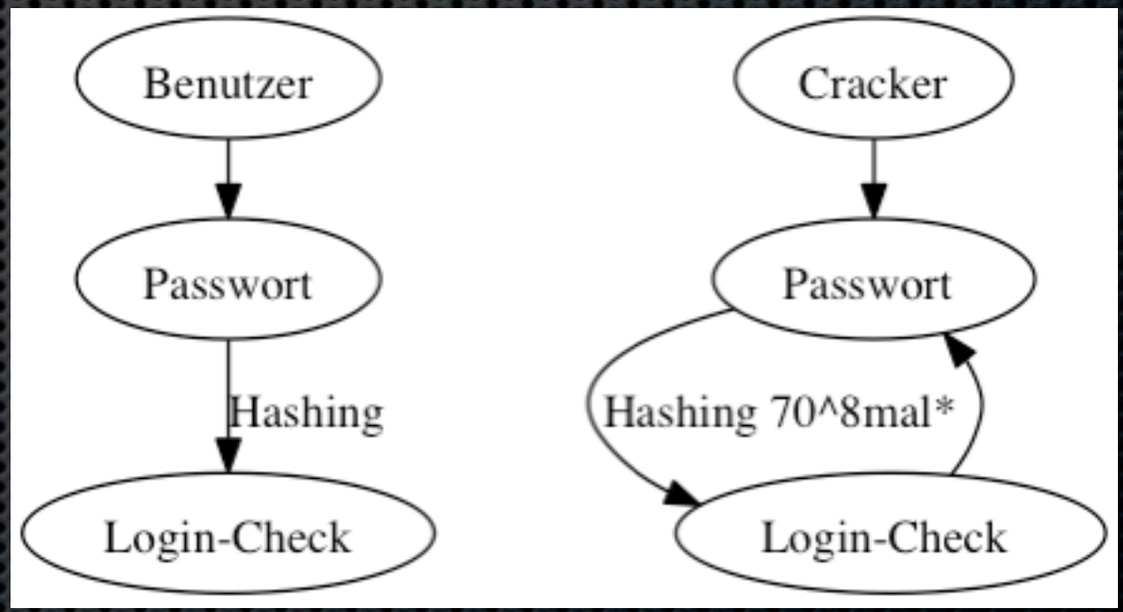


KDF: key derivation function

hashes are calculated fast

- one hashing per login
- per password:

  576.480.100.000.000
  ≈5,8*10^13 times hashing (tries)



*70^8 [A-Za-z0-9#+-_.,;:]{8}

KDF - more work for all

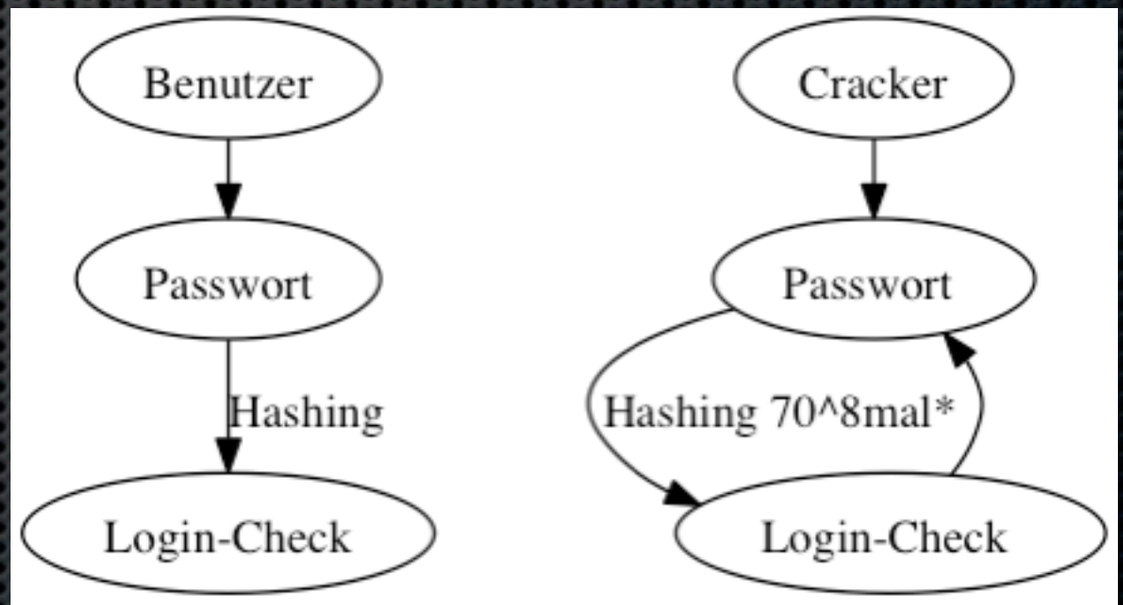
- hash more than one time (>1000 times)
- $n = 10^4; $res = „some String“;

```
for($i = 0;$i<=$n;$i++){
    $res = sha2($res);
}
return $res
```

hashes are calculated fast

- 10.000 times hashing per login
- per password

  5.764.801.000.000.000.000
  ≈5,8*10^17 times hashing



*70^8 [A-Za-z0-9#+-_.,;:]{8}

*10^4 times for KDF

Key derivation function

- abstract
- **hash** function **used often** on a **input string** and a **salt**
- example (WPA2 (PBKDF2))
- HMAC-SHA1
- 4096
- wifi key
- SSID

recommendations

- key derivation function, e.g.
  - **PBKDF2-HMAC-SHA256, c = 86000**
  - bcrypt, cost = 11
  - scrypt, N = 214,r = 8,p = 1
- long salt e.g.
  - md5(username)
  - md5(time())
  - random string

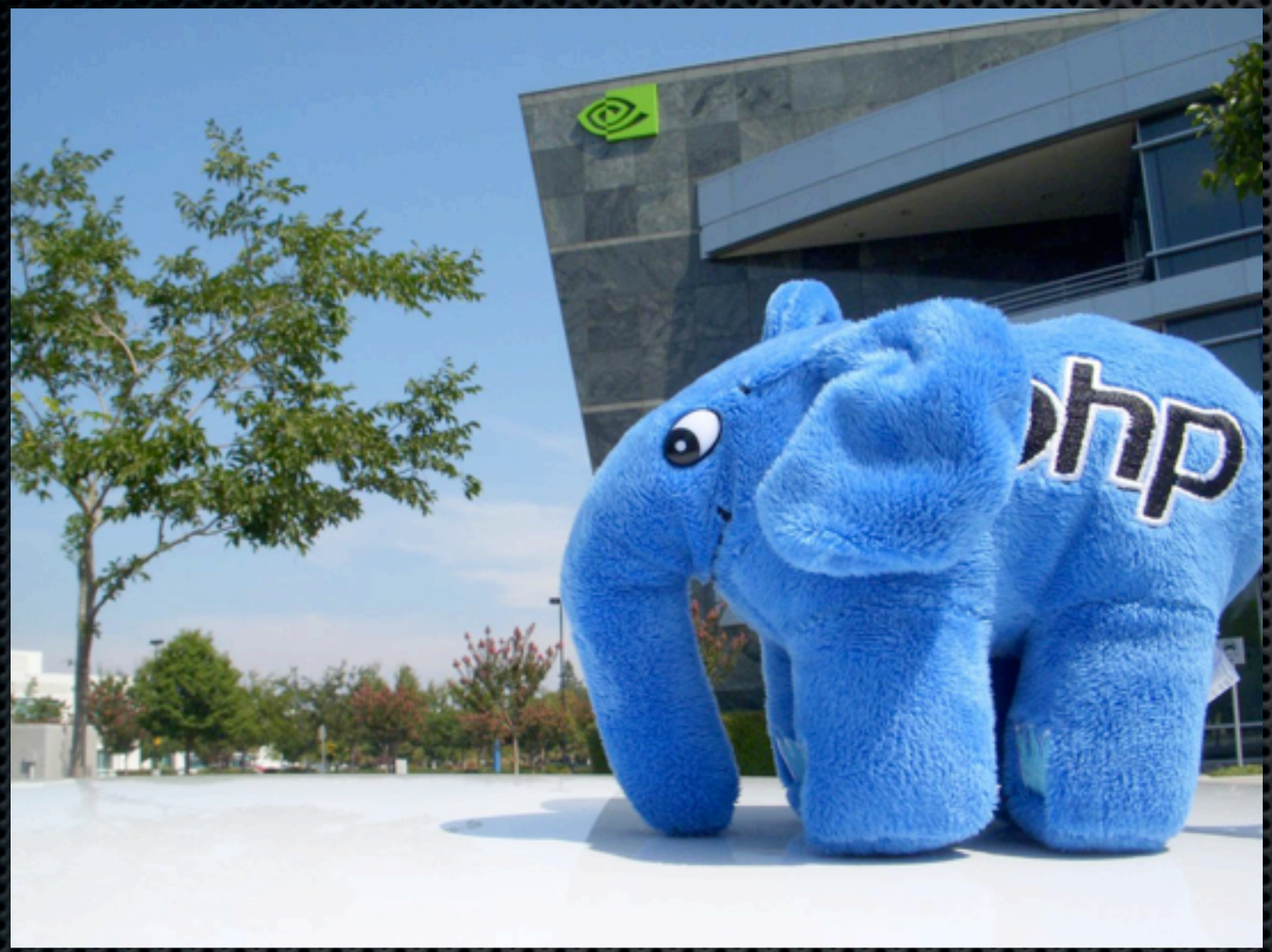
solution for magento

- Ikonoshirt_PBKDF2
- implements PBKDF2 for
  - customer passwords
  - admin user passwords
  - api passwords
- migrates passwords for all entities after login
- https://github.com/ikonoshirt/pbkdf2

don‘t forget your environment!

- HTTPS for users
  - and to the database (if needed)
- registration and password forgotten:
  - don‘t send passwords via mail
    - better: send link to reset the password (so the password is send via HTTPS!)

Questions?



- Questions? Comments?
- Presentation: http://www.ikonoshirt.de/stuff
- Me: http://www.fabian-blechschmidt.de

Magento™ Hackathon

Learn more about the Magento Hackathons and register to be part of a Magento Hackathon.

Register for the next Hackathon »

http://www.magento-hackathon.de

Munich (26. - 28.10.2012)

Price: 50 €

The 2nd Magento Hackathon Munich will take place at the office of Jarlssen, who hosted the first one in March 2012. http://bit.ly/6extensions_2days. Developers from all over Europe will join and share their experiences. Please note that this is a conference from developers for developers.

View details »

Berlin (11. - 13.1.2013)

Price: 120 €

The Magento Hackathon will take place in a nice youth hostel in the south of Berlin, Germany's capital city. An international meeting for new and nosey, as well as for experienced and committed Magento developers.

View details »

Sources

- http://mashable.com/2012/06/15/hard-to-hack-password/
- http://erratasec.blogspot.de/2012/06/linkedin-vs-password-cracking.html
- http://blog.zoller.lu/2012/06/storing-password-securely-hashses-salts.html
- http://de.wikipedia.org/wiki/Rainbow_Table
- http://en.wikipedia.org/wiki/PBKDF2
- http://www.openwall.com/presentations/PHDays2012-Password-Security/
- http://en.wikipedia.org/wiki/Bcrypt
- http://www.mscs.dal.ca/~selinger/md5collision/
- http://www.bsdcan.org/2009/schedule/attachments/86_scrypt_slides.pdf
- http://www.ietf.org/rfc/rfc2898.txt
- http://www.zyxist.com/en/archives/111 (not up to date. Salt + Multiple Hashing is good!)

Images

- Hashfunction: http://de.wikipedia.org/w/index.php?title=Datei:Hash_function_long.svg&filetimestamp=20051202013811
- Comic: http://xkcd.com/936/
- ElePHPant: http://www.flickr.com/photos/calevans/2856636492/in/pool-elephpants/