Cracking A5/1

Lucian Adrian Grijincu
lucian.grijincu@gmail.com

OSP

January 19, 2010

1. Introduction
2. Rainbow tables
3. Hadoop
4. Cracking A5/1 with Hadoop
5. Conclusion

1 Introduction

2 Rainbow tables

3 Hadoop

4 Cracking A5/1 with Hadoop

5 Conclusion

A5/what?

- A5/1 - stream cipher used for OTA privacy in GSM networks
- A5/2 - a weaker version of A5/1
- A5/3 - (aka KASUMI) newer version, other kind of algorithm

A5/1

- designed from the start to be easy to break:
- 1994 - first disclosure of the algorithm
- 1997 - A5/1 shown academically broken
- 2000 - more proof ...
- 2003 - more proof ...
- 2005 - and then some more ...
- 2008 - rainbow tables computed (but never released publicly)
- 2009 - A5/1 Security Project announce project to build public rainbow table
- 2010 - rainbow tables released on bittorrent (2TB)

A5/1 used in GSM

- first plain-text frames of a GSM call have a distinct pattern:
  - some bits are always zero
  - ACK bits
  - state encoding bits
- this limmits the search space significantly

History lesson

similar technique used to break the German cypher in WW2:

- messages longer than a page began with
- FORT (*Fortsetzung*)
- the time of the previous message between `Y`s
- the time of the previous message between `Y`s, again!
- "continuation of message sent at 2330" – "FORTYWEEPYYWEEPY"

1 Introduction

2 Rainbow tables

3 Hadoop

4 Cracking A5/1 with Hadoop

5 Conclusion

Cypher tables

- for each plain text
    - for each password
        - compute crypto(text, password)

Cypher tables

- pass=0000
  - 0000 - A7B7
  - 0001 - HJ89
  - ...
  - 9999 - 21J3
- pass=0001
  - 0000 - 32H4
  - 0001 - 5JL3
  - ...
  - 9999 - HJ89

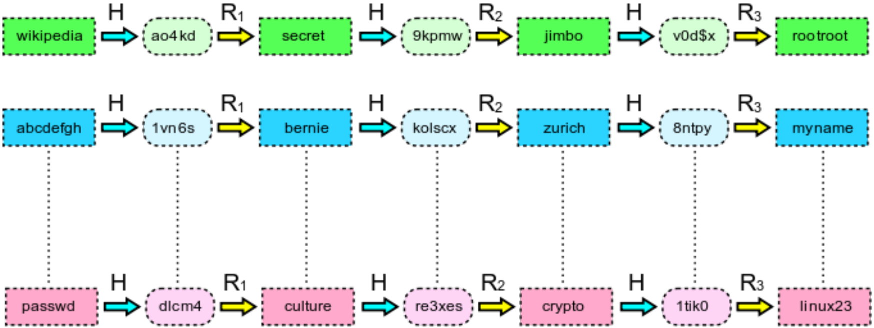
Cypher tables

- size grows exponentially with
  - plain text length
  - password length
- duplicates in the table. HJ89 bellongs to:
  - text=0001 and pass=0000
  - text=9999 and pass=0001
  - etc.

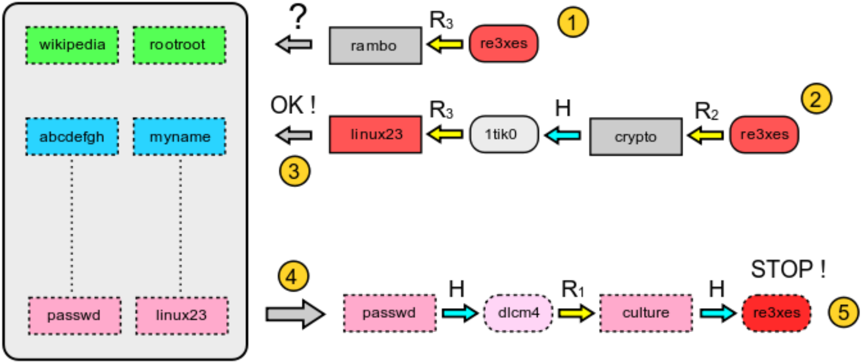
Rainbow tables

- select a random set of input secret values
- reduce the size of the table
- increase the lookup time

Rainbow tables



Rainbow tables



Rainbow tables

- R functions are not inverses of H!
- chains of 2^{15} R functions per table
- posibility of overlapping last entries:
- use many tables with other sets of R functions

1. Introduction
2. Rainbow tables
3. **Hadoop**
4. Cracking A5/1 with Hadoop
5. Conclusion

Hadoop

- open source map-reduce
- highly scalable (thousand of nodes)

Hadoop

Map

- read input
- create basic $< key, value >$ pairs

Hadoop

Reduce

- combine $< key, value >$ pairs with same *key*
- write output

1 Introduction

2 Rainbow tables

3 Hadoop

4 Cracking A5/1 with Hadoop

5 Conclusion

Cracking steps

precalculate tables - done once

1. create a set of random initial secret values
2. map-reduce the creation of the tables

search for a secret based on hashes

Table calculation - **Map**

- break input set of secrets
- each mapper computes a chain
- results are sent with
    - *key*=last secret in chain
    - *value*=first secret in chain

Table calculation - **Reduce**

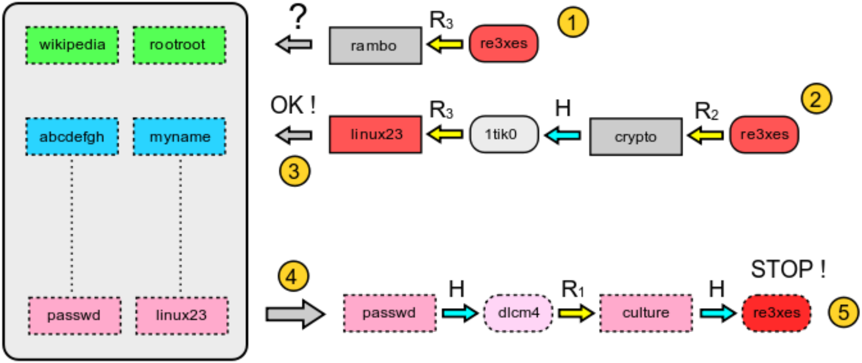
- reduce multiple $< key, value >$ pairs:
- group entries in tables
- group all start secrets that generate the same end secret

Lookup

in each table:

- **Map** - find all secrets that might generate the searched *hash*
- **Reduce** - from all secrets, only select the most frequent appearing secret

Lookup algorithm



Conclusion

- depending on the size of the chains: 1TB - 32TB tables
- this permits near real-time lookup

Other GSM bad news

- A5/2 - is weaker than A5/1
- key sizes less than 64 bits make cracking possible
- hardware and software (open source) for GSM radio transmissions is already avaliable
- A5/3 - has 64 and 128 bit key sizes
- devices that support A5/3 use 64 bits because it consumes less power

Why weak algorithms?

- they don't protect the user privacy
- only protect network operator's pockets
- crippled from the start to permit eavesdropping

Other results

- The C3 group used 40 NVIDIA CUDA machines for three months
- rainbow table size: 2TB
- efficient distribuition of this table permits real-time cracking if the call is intercepted from the start