Final Presentation

Rainbow Tables

Group 2
Cybersecurity

Speed Table Generation vs. Lookup

- Assumption:
  - Table Generation is slow
  - Lookup is fast
- But is it always?

- #chains = 100, Iterations = 100.000
- Generation: 7 seconds
- Lookup: aborted after 30 minutes

Speed Table Generation vs. Lookup

- Assume:
  - Reduction and Hash have equal execution time
  - n = Number of Iterations

- Table Generation: $\#chains * (2n-1)$ Operations
- Lookup: $\sum_{i=1}^{n} 2i-1$ Operations

- Table Generation: $\sim 2 * 10^7$ Operations
- Lookup: $\sim 10^{10}$ Operations

Search for the right k

- Task: find a k so that a Hash, reduced to its first k Bits lead to successful lookups

- $k = 16$, 1000 Iterations

```
$ python3 run.py
Finished filling 40000 rows in 39.583 seconds (parallel=True)
RainbowTable has 127 rows from 40000 input words
```

- $k = 24$, 1000 Iterations

```
$ python3 run.py
Finished filling 40000 rows in 49.276 seconds (parallel=True)
RainbowTable has 17545 rows from 40000 input words
```

Search for the right k

- Task: find a k so that a Hash, reduced to its first k Bits lead to successful lookups

- $k = 24$, 1000 Iterations

```
$ python3 run.py
Finished filling 40000 rows in 49.276 seconds (parallel=True)
RainbowTable has 17545 rows from 40000 input words
```

- $k = 32$, 1000 Iterations

```
$ python3 run.py
Finished filling 40000 rows in 57.979 seconds (parallel=True)
RainbowTable has 39820 rows from 40000 input words
```

Search for the right k

- Need at least 32-bit ⇒ $2^{32} = 4.29 * 10^9$ possible Hashes
- During development: 10000 iterations, 40161 words
- $10000 - 1\ \mathit{hashes}\ *\ 40161\ \mathit{chains}\ =\ 4.01 * 10^8\ \mathit{hashes}$
- Generation took about 10 minutes (8 Threads)

⇒ Need 10 times that! And also need time for lookup

Need for better performance

- Port of Python Implementation to Go
  - Slower ...
- Port of Python Implementation to Rust
  - Table generation and lookup twice as fast
  - With parallelization lookup time quartered

Search for the right parameters

- Lookup time is down to about 35 seconds for 10000 iterations
- Wordlist of 352000 words should be done in 44 minutes
- $10000-1\ hashes * 352000\ chains = 3.51*10^9\ hashes$

⇒ About 82% hit rate

Final Rainbow Table

- Adjusted rockyou.txt, 352000 words
- 248092 chains in 45 min
- 2,48*10^9 hashes ⇒ 57,8% hit chance

- No hit: hello, foo, bar, foobar, 1234567890, ~40s
- Hit:
  - h(abcdefgh) = 48bf2e86 -> ggipdarn, 15s
  - h(HalloWelt) = db968f3e -> mcnoqscf, 32s
  - h(password) = c3f84761 -> bzgnapee, 10s
  - h(Passwort) = 9da461ad -> uzfsbgbj, 0,1s

Thank you!