Team ThreadRippers
Presentation-2

Omkar Kakade
Indrajeet Vidhate
Password Cracking (using Rainbow Tables)
Summary

- Cannot store password in plain text.
- Cryptographic Hash are stored.
- Program to crack hashes using rainbow tables.
- Rainbow Table is a precomputed table of cryptographic hashes.

Why?

- Practical example of space - time tradeoff. (less computer processing time and more storage)
Simplified rainbow table with 3 reduction functions
Paper 1 - Citations

- Title - An improved parallel implementation of Rainbow Crack using MPI
- Author - Edward R. Sykes, Wesley Skoczen
- Name of Journal - Journal of Computational Science
- Date (Published) - 1 November 2013
- Pages - 536 - 541
Paper 1 - Problem Addressed

- Reducing precomputation time (Time required to construct the rainbow table) using parallel paradigms.
- Reducing time to crack that hash by using parallel programming optimizations.
Paper 1 - Novel Contributions

- Come up with parallel implementation for Rainbow table based cracking which didn’t exist earlier.

- Individual chains can be parallelized since there is no inter-process communication and they can be computed independently.

- Earlier = first value of chain generated -> disk | final value generated -> disk.

- 3 models for handling file writing and comparison using run times to finalize the best.
  - Multiple processors, each process -> separate files (using native file library)
  - Multiple processors, each process -> separate files (using MPI file library functions)
  - Multiple processors -> one file (using MPI file library functions)
Paper 1 - Novel Contributions

- Reduced a problem that could take 6 years to a few days.

| Table 1 |
|-----------------|-----------------|
| **Results of tests runs of RainbowCrack on an AMD athlon 2.0 GHz desktop machine and our parallel implementation of RainbowCrack on a SHARCNET supercomputer using 16, 32, 64, and 90 processors.** |

<table>
<thead>
<tr>
<th></th>
<th>Single processor</th>
<th>MPI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Table time creation</td>
<td>Cracking</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alpha</td>
<td>2 days 18 h</td>
<td>3-30 s</td>
</tr>
<tr>
<td>Alpha-numeric</td>
<td>15 days 17 h</td>
<td>6-65 s</td>
</tr>
<tr>
<td>Alpha-numeric-symbol14</td>
<td>224 days</td>
<td>28-287 s</td>
</tr>
<tr>
<td>(7 months)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alpha-numeric-all characters</td>
<td>2354 days</td>
<td>197-915 s</td>
</tr>
<tr>
<td>(6 years !)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Paper 1 - Usage in our Project

- Parallelize individual chain generation since it can be computed independently.
- Chain writing can be handled by using per thread reduction variable which will yield better performance due to eliminating communication overhead.
- Strong scaling findings.
Paper 2 - Citations

- **Title**: Rainbow Table optimization for Password recovery
- **Author**: Vrizlynn L. L. Thing, Hwei-Ming Ying
- **Name of Journal**: International Journal on Advances in Software, Vol. 4 number 3 & 4
- **Date**: 2011
- **Pages**: 479 - 488
- **URL**: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.674.1355&rep=rep1&type=pdf
Simplified rainbow table with 3 reduction functions
Paper 2 - Problem Addressed

- Achieving storage conservation as compared to initial approach suggested for Rainbow-Tables by P. Oechlin
Paper 2 - Novel Contributions

- A new approach of chain generation for reduced chain - endpoint mappings
- Both uses same number of reduction functions but the approach suggested in given paper proposes one-to-many mappings in hash-reduction chains
- The approach computes hash for a particular starting plain text and then takes into account decimal representation for the same (‘H’)
- Then to compute \((H+1) \mod 2^j\) upto \((H+K) \mod 2^j\), where \(j\) is the number of bits in particular hash (Eg. \(j = 128\) for MD5)
- Then we calculate normal hash-reduce chains starting from this above obtained branches
- All the set of plain texts obtained will be represented by a single plaintext
Paper 2 - Usage in our Project

- We can assign different size chunks of starting plain texts to different threads in Parallel. For
  As there is no sequential dependency in parallel chains
- The technique for chain generation suggested will reduce the probability of collision and
  subsequent merging of chains
Paper 3 - Citations

- Title: Heterogeneous Rainbow Table Widths Provide Faster Cryptanalyses
- Author: Gildas Avoine, Xavier Carpent
- Name of Journal: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security
- Date: 2 April 2017
- Pages: 815-822
- URL: http://dx.doi.org/10.1145/3052973.3053030
The rainbow-tables are not well utilized if all tables have same width of chains i.e. equal number of chains with equal number of columns.

Such consideration can introduce performance problems if sequential search is applied.
Heterogenous Table -

1. Use of new table structures compared to traditional table structures.
2. Heterogenous Table structure permits to have multiple rainbow tables of different lengths.
3. Heterogeneous tables are 40% faster than homogeneous tables.
Figure 3: Shapes of a set of 4 homogeneous tables (top row) compared to optimal heterogeneous tables (bottom row) using the same memory on a set of size $N = 2^{40}$. The vertical scale (number of chains) is 10000 smaller than the horizontal scale (chain lengths).
In case of rainbow table with heterogeneous length, traditional sequential table search approach is not optimal for performance compared to performing parallel search on such tables.

As our algorithm, is searching the tables parallely it can be benefit from the heterogeneous table structure.
THANK YOU
Q&A