Cracking Passwords with Time-memory Trade-offs

Gildas Avoine
INSA Rennes (France), UCL (Belgium)
SUMMARY

Motivations
Hellman Tables
Oechslin Tables
Real Life Examples
Rainbow Tables with Fingerprints
Conclusion
MOTIVATIONS

- Motivations
  - Hellman Tables
  - Oechslin Tables
  - Real Life Examples
  - Rainbow Tables with Fingerprints
  - Conclusion
Function $h : A \rightarrow B$ that is easy to compute on every input, but hard to invert given the image of an arbitrary input.
Example: Password-based Authentication

User (username, pwd) \[\rightarrow\] Computer

\[\text{username, pwd} \rightarrow \text{Compute } h(pwd)\]

Gildas Avoine
Exhaustive Search

- **Online exhaustive search:**
  - Computation: $N := |A|$
  - Storage: 0
  - Precalculation: 0

- **Precalculated exhaustive search:**
  - Computation: 0
  - Storage: $N$
  - Precalculation: $N$
HELMAN TABLES

- Motivations
- Hellman Tables
- Oechslin Tables
- Real Life Examples
- Rainbow Tables with Fingerprints
- Conclusion
Precalculation Phase

- Precalculation phase to speed up the online attack: \[ T \propto \frac{N^2}{M^2} \]
Reduction Functions

- $R : B \rightarrow A$ is used to map a point from $B$ to $A$ \textit{arbitrarily.}

- It should be \textbf{fast} to compute (w.r.t. $h$)

- $R$ should be \textbf{surjective}.

- $R$ should be \textbf{deterministic}.

- $\forall a \in A, \ |R^{-1}(a)| \approx \frac{|B|}{|A|}$

- Typically, $R : b \mapsto b \mod N$. 
Precalculation Phase (recap)

- Invert \( h : A \rightarrow B \).
- Define \( R : B \rightarrow A \) an arbitrary (reduction) function.
- Define \( f : A \rightarrow A \) such that \( f = R \circ h \).
- Chains are generated from arbitrary values in \( A \).

\[
\begin{align*}
S_1 &= X_{1,1} \xrightarrow{f} X_{1,2} \xrightarrow{f} X_{1,3} \xrightarrow{f} \ldots \xrightarrow{f} X_{1,t} = E_1 \\
S_2 &= X_{2,1} \xrightarrow{f} X_{2,2} \xrightarrow{f} X_{2,3} \xrightarrow{f} \ldots \xrightarrow{f} X_{2,t} = E_2 \\
&
\vdots
\end{align*}
\]

\[
\begin{align*}
S_m &= X_{m,1} \xrightarrow{f} X_{m,2} \xrightarrow{f} X_{m,3} \xrightarrow{f} \ldots \xrightarrow{f} X_{m,t} = E_m \\
&
\vdots
\end{align*}
\]

- The generated values should cover the set \( A \) (probabilistic).
- Only the first and the last element of each chain is stored.
Online Attack

\[ h: A \rightarrow B \]
\[ R: B \rightarrow A \]

*R is the reduction function*
Given one output $y \in B$, we compute $y_1 := R(y)$ and generate a chain starting at $y_1$: $y_1 \xrightarrow{f} y_2 \xrightarrow{f} y_3 \xrightarrow{f} \ldots y_s$.
Coverage and Collisions

- **Collisions** occur during the precalculation phase.
- **Several tables** with different reduction functions.
OECHSLIN TABLES

- Motivations
- Hellman Tables
- Oechslin Tables
- Real Life Examples
- Rainbow Tables with Fingerprints
- Conclusion
Using Several Reduction Functions (Oechslin, 2003)

- Use a different reduction function per column: rainbow tables.
- Invert $h : A \rightarrow B$.
- Define $R_i : B \rightarrow A$ arbitrary (reduction) functions.
- Define $f_i : A \rightarrow A$ such that $f_i = R_i \circ h$.

\[
S_1 = X_{1,1} \xrightarrow{f_1} X_{1,2} \xrightarrow{f_3} X_{1,3} \xrightarrow{f_3} \ldots \xrightarrow{f_3} X_{1,t} = E_1
\]

\[
S_2 = X_{2,1} \xrightarrow{f_1} X_{2,2} \xrightarrow{f_3} X_{2,3} \xrightarrow{f_3} \ldots \xrightarrow{f_3} X_{2,t} = E_2
\]

\[
\vdots
\]

\[
S_m = X_{m,1} \xrightarrow{f_1} X_{m,2} \xrightarrow{f_3} X_{m,3} \xrightarrow{f_3} \ldots \xrightarrow{f_3} X_{m,t} = E_m
\]
Discarding the Merges

- If 2 chains collide in different columns, they don’t merge.
- If 2 chains collide in same column, merge can be detected.

A table without merges is said **perfect** (*clean*).
Given one output $y \in B$, we compute $y_1 := R(y)$ and generate a chain starting at $y_1$:

$$y_1 \xrightarrow{f_{t-s}} y_2 \xrightarrow{f_{t-s+1}} y_3 \xrightarrow{f_{t-s+2}} \ldots y_s$$
Success Probability of a Table is Bounded

Theorem

Given \( t \) and a sufficiently large \( N \), the expected maximum number of chains per perfect rainbow table without merge is:

\[
m_{\text{max}}(t) \approx \frac{2N}{t + 1}.
\]

Theorem

Given \( t \), for any problem of size \( N \), the expected maximum probability of success of a single perfect rainbow table is:

\[
P_{\text{max}}(t) \approx 1 - \left(1 - \frac{2}{t + 1}\right)^t
\]

which tends toward \( 1 - e^{-2} \approx 86\% \) when \( t \) is large.
Average Cryptanalysis Time

**Theorem**

Given $N$, $m$, $\ell$, and $t$, the average cryptanalysis time is:

$$T = \sum_{c=t-\left\lfloor \frac{k-1}{\ell} \right\rfloor}^{k=\ell t} p_k \left( \frac{(t-c)(t-c+1)}{2} \right) + \sum_{i=c}^{i=t} q_i i \ell + \left(1 - \frac{m}{N}\right)^{\ell t} \left( \frac{t(t-1)}{2} \right) + \sum_{i=1}^{i=t} q_i i \ell$$

where

$$q_i = 1 - \frac{m}{N} - \frac{i(i-1)}{t(t+1)}.$$
REAL LIFE EXAMPLES

- Motivations
- Hellman Tables
- Oechslin Tables
- Real Life Examples
- Rainbow Tables with Fingerprints
- Conclusion
Win98/ME/2k/XP uses the Lan Manager Hash (LM hash).
- The password is cut in two blocks of 7 characters.
- Lowercase letters are converted to uppercase. Not salted.
Windows LM Hash (Results)

Cracking an **alphanumeric password** (LM Hash) on a PC. Size of the problem: \( N = 8.06 \times 10^{10} = 2^{36.23} \).

<table>
<thead>
<tr>
<th></th>
<th>Brute Force</th>
<th>TMTO</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Online Attack (op)</strong></td>
<td>( 4.03 \times 10^{10} )</td>
<td>( 1.13 \times 10^{6} )</td>
</tr>
<tr>
<td>Time</td>
<td>2 h 15</td>
<td>0.226 sec</td>
</tr>
<tr>
<td><strong>Precalculation (op)</strong></td>
<td>0</td>
<td>( 1.42 \times 10^{13} )</td>
</tr>
<tr>
<td>Time</td>
<td>0</td>
<td>33 days</td>
</tr>
<tr>
<td>Storage</td>
<td>0</td>
<td>2 GB</td>
</tr>
</tbody>
</table>
## Statistics from 10,000 Leaked Hotmail Passwords

<table>
<thead>
<tr>
<th>Password Type</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>numeric</td>
<td>19%</td>
</tr>
<tr>
<td>lower case alpha</td>
<td>42%</td>
</tr>
<tr>
<td>mixed case alpha</td>
<td>3%</td>
</tr>
<tr>
<td>mixed numeric alpha</td>
<td>30%</td>
</tr>
<tr>
<td>other charac</td>
<td>6%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Password Length</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>≤ 7</td>
<td>37%</td>
</tr>
<tr>
<td>≤ 8</td>
<td>58%</td>
</tr>
<tr>
<td>≤ 9</td>
<td>70%</td>
</tr>
</tbody>
</table>
Windows NT LM Passwords

- Win NT/2000/XP/Vista/Seven uses the **NT LM Hash**.
- The password is **no longer cut** in two blocks.
- Lowercase letters are **not converted** to uppercase. **Not salted**.
Cracking a 7-char (max) alphanumerical password (NT LM Hash) on a PC. Size of the problem: $N = 2^{41.7}$.

<table>
<thead>
<tr>
<th></th>
<th>Brute Force</th>
<th>TMTO</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Online Attack (op)</strong></td>
<td>1.78 $\times 10^{12}$</td>
<td>4.48 $\times 10^7$</td>
</tr>
<tr>
<td>Time</td>
<td>99 hrs</td>
<td>9.0 sec</td>
</tr>
<tr>
<td><strong>Precalculation (op)</strong></td>
<td>0</td>
<td>6.29 $\times 10^{14}$</td>
</tr>
<tr>
<td>Time</td>
<td>0</td>
<td>1458 days</td>
</tr>
<tr>
<td>Storage</td>
<td>0</td>
<td>16 GB</td>
</tr>
</tbody>
</table>
RAINBOW TABLES WITH FINGERPRINTS

(Joint work with A. Bourgeois and X. Carpent)

- Motivations
- Hellman Tables
- Oechslin Tables
- Real Life Examples
- Rainbow Tables with Fingerprints
- Conclusion
Given one output $y \in B$, we compute $y_1 := R(y)$ and generate a chain starting at $y_1$:

$$
y_1 \xrightarrow{f_{t-s}} y_2 \xrightarrow{f_{t-s+1}} y_3 \xrightarrow{f_{t-s+2}} \ldots y_s
$$

The diagram illustrates the process with $S_1 \ldots S_m$ and $E_1 \ldots E_m$ as the source and endpoint sets, respectively. The checkpoint time needed to rebuild the chain is indicated, along with the time needed to find a matching endpoint.
Endpoints and checkpoints share the same nature.

Each column contains a ridge (potentially empty).

A fingerprint is a series of ridges for a given chain.

Fingerprints are stored instead of the endpoints.

We look for matching fingerprints (instead of endpoints).
Ridges and Fingerprints

time needed to rebuild the chain

time needed to find a matching endpoint
Theorem

The average amount of evaluations of $h$ during the online phase using the rainbow tables with fingerprints is:

$$T = \sum_{k=1}^{\ell t} \frac{m}{N} \left( 1 - \frac{m}{N} \right)^{k-1} (W_k + Q_k) + \left( 1 - \frac{m}{N} \right)^{\ell t} (W_{\ell t} + Q_{\ell t}),$$

$$c_i = t - \left\lceil \frac{i - 1}{\ell} \right\rceil,$$

$$q_c = 1 - \prod_{i=c}^{t} \left( 1 - \frac{m_i}{N} \right),$$

$$W_k = \sum_{i=1}^{k} (t - c_i),$$

$$P_c = \sum_{i=c}^{t} \left[ \prod_{j=c}^{i-1} \phi_j \right] (q_i - q_{i+1}),$$

$$Q_k = \sum_{i=1}^{k} (c_i - 1)(P_{c_i} + E_{c_i}),$$

$$E_c = (m - q_c) \prod_{i=c}^{t} \phi_i.$$
Cracking a 7-char (max) alphanumerical password (NT LM Hash) on a PC. Size of the problem: \( N = 2^{41.7} \).

<table>
<thead>
<tr>
<th></th>
<th>Brute Force</th>
<th>TMTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Attack (op)</td>
<td>1.78 \times 10^{12}</td>
<td>2.94 \times 10^7</td>
</tr>
<tr>
<td>Time</td>
<td>99 hrs</td>
<td>5.9 sec</td>
</tr>
<tr>
<td>Precalculation (op)</td>
<td>0</td>
<td>6.29 \times 10^{14}</td>
</tr>
<tr>
<td>Time</td>
<td>0</td>
<td>1458 days</td>
</tr>
<tr>
<td>Storage</td>
<td>0</td>
<td>16 GB</td>
</tr>
</tbody>
</table>
CONCLUSION

- Motivations
- Hellman Tables
- Oechslin Tables
- Real Life Examples
- Rainbow Tables with Fingerprints
- Conclusion
A TMTO is never better than a brute force.

TMTO makes sense in several scenarios.
- Attack repeated several times.
- Lunchtime attack.
- Attacker is not powerful but can download tables.

Two conditions to perform a TMTO.
- Reasonably-sized problem.
- One-way function (or chosen plaintext attack on a ciphertext).

Rainbow tables with fingerprints are a new view of rainbow tables.