Time-memory Trade-offs Applied to Non-uniform Distributions

Gildas Avoine

INSA Rennes (France)

Joint work with Xavier Carpent and Cédric Lauradoux

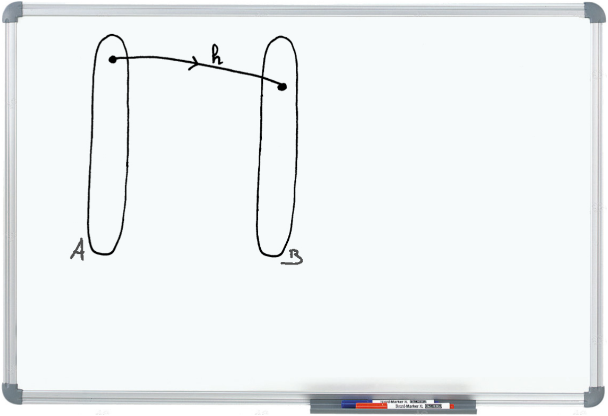
SUMMARY

- Motivations
- Hellman's TMTO
- Real Life Example
- Interleaved TMTOs
- Conclusion

MOTIVATIONS

One-way Function

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)		Computer
	username, pwd $\longrightarrow$	
		Compute $h(\text{pwd})$

username_1	$h(\text{pwd}_1)$
username_2	$h(\text{pwd}_2)$
username_3	$h(\text{pwd}_3)$
$\vdots$	$\vdots$
username_N	$h(\text{pwd}_N)$

Exhaustive Search

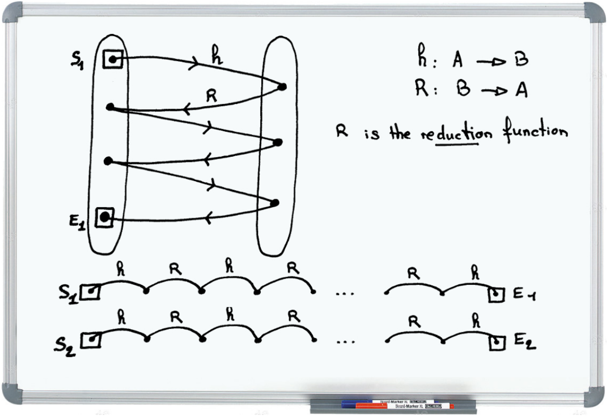
- Online exhaustive search:
  - Computation: $N := |A|$
  - Storage: 0
  - Precalculation: 0

- Precalculated exhaustive search:
  - Computation: 0
  - Storage: N
  - Precalculation: N

HELLMAN'S TMTO

Precalculation Phase

- Martin Hellman's cryptanalytic time-memory trade-off (1980).
- Precalculation phase to speed up the online attack: $T \propto \frac{N^2}{M^2}$

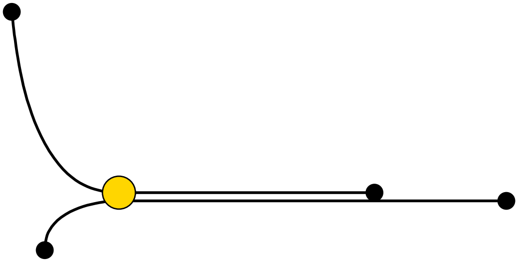


Reduction Functions

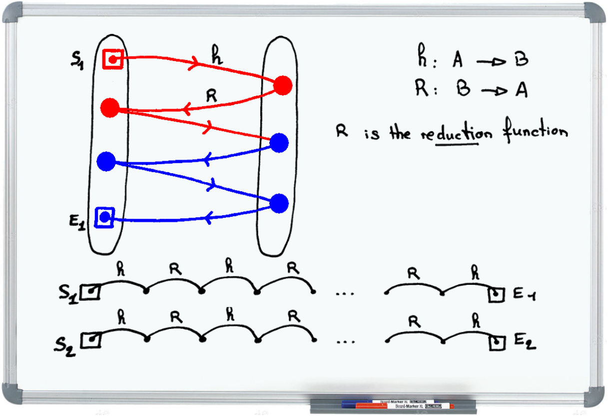
- $R : B \to A$ is used to map a point from B to A arbitrarily
- It should be fast to compute (w.r.t. h)
- R should be surjective.
- R should be deterministic.
- $\forall a \in A, \;\; |R^{-1}(a)| \approx \frac{|B|}{|A|}$
- Typically, $R : b \mapsto b \bmod N$.

Coverage and Collisions

- Collisions occur during the precalculation phase.
- Many tables with different reduction functions.

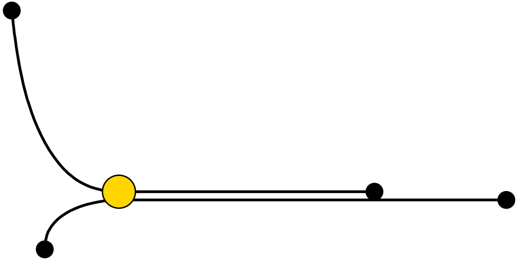


Online Attack



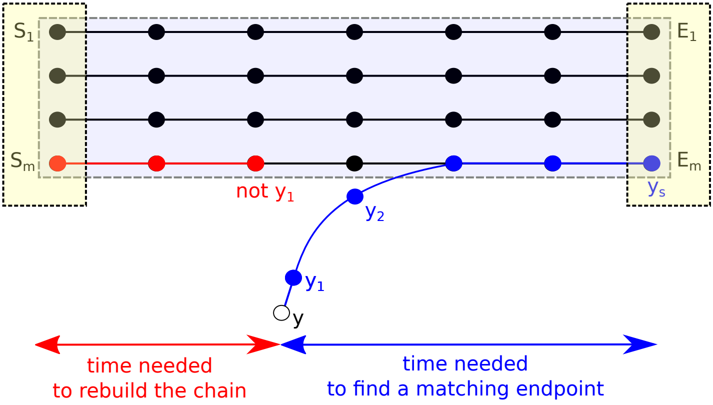
Collisions during the Online Phase

- Collisions occur between online chain and precalculated ones.



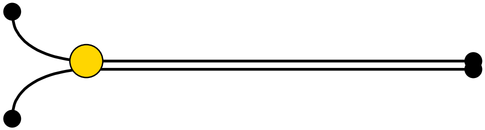
Online Attack (Recap)

- Given one output $y \in B$, we compute $y_1 := R(y)$ and generate a chain starting at y_1: $y_1 \xrightarrow{f} y_2 \xrightarrow{f} y_3 \xrightarrow{f} \ldots y_s$



Oechslin's Rainbow Tables (2003)

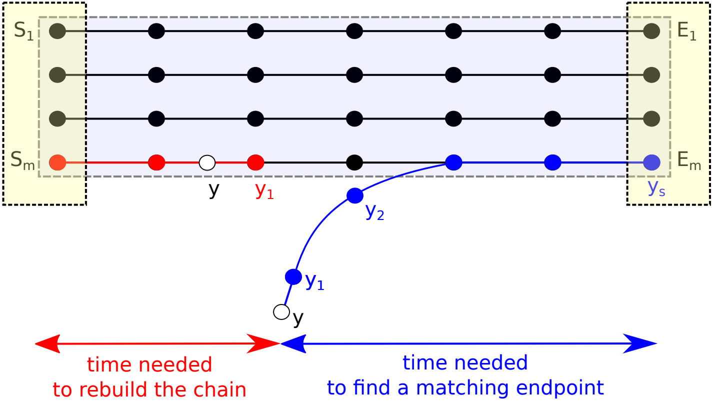
- Use a different reduction function per column: rainbow tables.
- Invert $h : A \to B$.
- Define $R_i : B \to A$ arbitrary (reduction) functions.
- If 2 chains collide in different columns, they don't merge.
- If 2 chains collide in same column, merge can be detected.



Online Procedure is More Complex

Given one output $y \in B$, we compute $y_1 := R(y)$ and generate a chain starting at y_1:

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



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, ℓ, and t, the average cryptanalysis time is:

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

$$\left(1-\frac{m}{N}\right)^{\ell t}\left(\frac{t(t-1)}{2} + \sum_{i=1}^{i=t} q_i i\right)\ell$$

where

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

REAL LIFE EXAMPLE

- Motivations
- Hellman's TMTO
- Real Life Example
- Interleaved TMTOs
- Conclusion

Windows NT LM Hash (Results)

Cracking a 7-char (max) alphanumerical password (NT LM Hash) on a PC. Size of the problem: $N = 2^{41.7}$.

	Brute Force	TMTO
Online Attack (op)	1.78×10^{12}	4.48×10^{7}
Time	99 hrs	9.0 sec
Precalculation (op)	0	6.29×10^{14}
Time	0	1458 days
Storage	0	16 GB

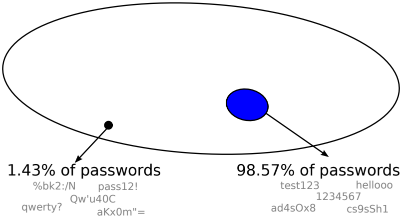
INTERLEAVED TMTOS

Interleaving Rational

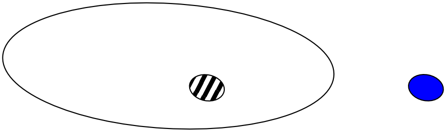
- A TMTO treats all possible preimages equally.
- What if preimages have a non-uniform distribution?
- Typical use case: passwords

Charset	Set Size	Proportion
Alphanum (length 1-7)	4.31×10^{12}	98.57%
AN + 34 special char. (length 7)	7.16×10^{13}	1.43%

Source: statistics on the RockYou dataset

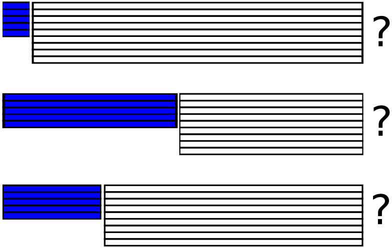


Interleaving Concept



Input space is partitionedA TMTO is built for each subspaceSequential search may be fine but is not the best solutionInstead, order of search is interleavedInterleaving order is computed such that it minimizes average time

Interleaving Memory Division



How to divide the memory between sub-TMTO's ? Grid search or metaheuristic search for the average time In this case: speedup of 16.45 w.r.t. single TMTO

CONCLUSION

Limits and Strength of TMTOs

- 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 equivalent problem).
- Interleaving is efficient when considering a non-uniform distribution: cracking passwords, deanonymization (hashed email or mac address).