Journal of Computing and Security

July 2016, Volume 3, Number 3 (pp. 155–162)

http://www.jcomsec.org

Expected Coverage of Perfect Chains in the Hellman Time Memory Trade-Off

Nasser Hossein Gharavi [a,*], Abodorasoul Mirghadri [a], Mohammad Abdollahi Azgomi [b], Sayyed Ahmad Mousavi [c]

[a] *Faculty of Communication and Information Technology, Imam Hossein University (IHU), Tehran, Iran.*
[b] *Iran University of Science and Technology (IUST), Iran.*
[c] *Shahid Bahonar University of Kerman, Kerman, Iran.*

ARTICLE INFO.

Article history:
Received: 18 January 2017
Revised: 16 September 2017
Accepted: 16 November 2017
Published Online: 10 February 2018

Keywords:
Time Memory Trade-Off, Perfect Chains, Random Mapping, Expected Value.

ABSTRACT

Critical overlap situations in the classical Hellman's cryptanalytic time memory trade-off method can be avoided, provided that during the precomputation phase, we generate perfect chains which are merge-free and loop-free chains. In this paper, we present asymptotic behavior of perfect chains in terms of time memory trade-off attacks. More precisely, we obtained expected values and variances for the coverage of perfect chains. We have also confirmed our theoretic outcomes with test results.

1 Introduction

Cryptanalytic time memory trade-offs (TMTO) have been introduced in 1980 by Hellman [1]. He tried to find an optimal solution between two extremes in searching of the large search spaces: the time-consuming exhaustive searching and the memory-consuming lookup tables. In its more general form, TMTO can be viewed as a general one-way function inverter. Let $\mathcal{X}$ be a search space of size N and $f : \mathcal{X} \to \mathcal{X}$ be a one-way function, *i.e.*, a function which is straightforward to compute, but which will be hard to invert. An example of interest is the— the function mapping a secret key to the encryption of a specific fixed known plaintext. TMTO is universal (independent of the type of function f) method for searching the solutions of the equation $f(x) = y$, for a fixed image $y \in \mathcal{X}$.

In precomputation phase (offline phase) of the Hellman's method, we generate m chains of $t + 1$ points from the search space $\mathcal{X}$ following a simple rule that in each chain, image of preceding point under the function f is the next point. In the attack phase (online phase), these chains can be used for searching the solutions of the equation $f(x) = y$, for a given image point $y \in \mathcal{X}$. This is accomplished by determining a specific chain that possibly includes y and searching through this chain. Consequently, the number of chains (m), as well as their lengths (t) are mostly selected in such a way that the number of points inside chains is approximately equal to N (the search space size).

In real situations, Hellman's TMTO technique suffers from several really serious problems. One of the main problems is the fact that the number of points could be repeated in one particular chain or in different chains, due to some unusual cases such as looping and merging of chains, is too high. Therefore, numerous occurrences of points lead us to a reduced coverage of the search space. Due to the fact that using TMTO approach we can discover just the points existing in chains, this technique is probabilistic. Hence a lower coverage of the search space immediately reduces the

* Corresponding author.
Email addresses: hgharavi@ihu.ac.ir (N.H. Gharavi), amrghdri@ihu.ac.ir (A. Mirghadri), azgomi@iust.ac.ir (M. Abdollahi Azgomi), s.a.mousavi@math.uk.ac.ir (S.A. Mousavi)
ISSN: 2322-4460

success of the attack.

Two basic improvements of the Hellman's method, that we are not treating in this work, are Distinguished Points (DP) method, which is attributed to Rivest in [2] and Rainbow method [3], announced by Oechslin; for more details regarding these methods see [4, 5].

For constructing an efficient TMTO table, we must confirm that the usage of our memory is certainly efficient. Consequently, in the precomputation phase, we should avoid storing repeated part of the chains, which merged with previously recorded chains or looped with itself. In DP and Rainbow methods, generally, more chains than really required are produced and then chains with similar ending points are removed. The resulting tables are called *perfect tables* [4–8]. In fact, perfect tables are tables in which merges and loops are occurred rarely.

Unlike DP and Rainbow methods, the removal of redundancies in classical Hellman's method cannot be done easily. In this paper, we will consider a generic procedure, introduced in the papers [6, 8], for constructing perfect version of the Hellman tables in perecomputation phase. During this technique, whenever a loop within a chain is found, construction of the chain might be terminated. Also, when a chain is merged with a previously constructed chain, the redundant part of the later chain will be omitted. This procedure guarantees unique elements in the table, *i.e.*, it leads us to perfect chains.

The main disadvantage of this approach, is that every single element of each chain that is created must looked up in all of the previously produced chains. The authors of the paper [8] offered to track the record of points previously included within the chains throughout the offline phase. On the other hand, these look ups can be performed efficiently using some high-speed cycle detection and merge detection algorithms [9–11].

In the perfect type of the TMTO techniques, characteristics of chains (for example the average lengths, coverage, and variances) shed more light on the perfect chain creation procedure, as well as, make it possible for getting a much better understanding of the TMTO. The performances of the perfect version of the DP and Rainbow methods are completely analyzed earlier and we are really not planning to discuss about them; for example you can see [4, 5, 12]. Additionally, there are several analysis for perfect chains behavior in the classical Hellman's method [6, 8], however at this point there isn't any explicit expression for the expected values and variances of the coverage of perfect chains in the classical Hellman's method. In this paper, we are going to obtain asymptotic formulas for expected values, and variances of the coverage of perfect chains in classical Hellman's method.

In the next section (Section 2), the basic Hellman time memory trade-off, its limitations and the perfect version of this method are described. In the Section 3, an asymptotic probabilistic analysis of the perfect chains is presented. More explicitly, expected values and variances for the coverage of perfect chains are obtained. Section 4 is devoted to experimental results of our theoretical formulas. We show experimentally that the theoretical results match practical outcomes. Finally, Section 5 provides a short summary and overview of the future works on this problem.

In the last of this section, note that, nevertheless in TMTO procedures we make use of a particular one-way function f, we can assume that f is a random mapping. This assumption is done by any theoretic analysis of trade-off algorithms. In very rough terms, a random mapping $f : \mathcal{X} \to \mathcal{X}$ is a function that assigns independent and random values $f(x) \in \mathcal{X}$ to each of its arguments $x \in \mathcal{X}$ [13, 14].

2 Hellman's Method and Its Perfect Version

In the Hellman's TMTO method, starting from a random point $x_0 \in \mathcal{X}$, iteratively evaluating f, a chain of length t is generated as follows:

$$x_0 \xrightarrow{f} x_1 = f(x_0) \xrightarrow{f} x_2 = f^2(x_0) \xrightarrow{f} \cdots \xrightarrow{f} x_t = f^t(x_0).$$

Let positive integers m and t that satisfy the relation $mt^2 \approx N$ (*matrix stopping rule*) be fixed. To construct a *Hellman matrix* of size $m \times (t+1)$, picking m random start points, chains of length t are created. In order to save memory, just the first and last columns of the Hellman matrix are kept in a table, known as *Hellman table*. Additionally, the table is sorted with respect to end points for speeding up the lookups in attack phase; for more details see [1, 4, 5].

$$\begin{array}{l}
SP_1 = x_{10} \xrightarrow{f} x_{11} \xrightarrow{f} x_{12} \xrightarrow{f} x_{13} \xrightarrow{f} \cdots \xrightarrow{f} x_{1(t-1)} \xrightarrow{f} x_{1t} = EP_1 \\
SP_2 = x_{20} \xrightarrow{f} x_{21} \xrightarrow{f} x_{22} \xrightarrow{f} x_{23} \xrightarrow{f} \cdots \xrightarrow{f} x_{2(t-1)} \xrightarrow{f} x_{2t} = EP_2 \\
SP_3 = x_{30} \xrightarrow{f} x_{31} \xrightarrow{f} x_{32} \xrightarrow{f} x_{33} \xrightarrow{f} \cdots \xrightarrow{f} x_{3(t-1)} \xrightarrow{f} x_{3t} = EP_3 \\
\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
SP_m = x_{m0} \xrightarrow{f} x_{m1} \xrightarrow{f} x_{m2} \xrightarrow{f} x_{m3} \xrightarrow{f} \cdots \xrightarrow{f} x_{m(t-1)} \xrightarrow{f} x_{mt} = EP_m
\end{array}$$

In the Hellman's method, one will try to be able to keep as many different points as possible by generating as long chains as possible; but various critical overlap scenarios can appear and increase constraints of the method.

(1) A chain may cycle. This is the situation where we discover $0 \le j_1 < j_2 \le t$ such that $x_{ij_1} = x_{ij_2}$, for some $1 \le i \le m$ (Figure 1 (a)).

(2) Two chains can merge. This is the case where there exist $i_1 \neq i_2$ such that $x_{i_1 j_1} = x_{i_2 j_2}$, for some j_1 and j_2. Which means that from the moment of the merge until the end of at least one chain, both chains include exactly the same points (Figure 1 (b)).

Consequently, multiple occurrences of points lead to a lower coverage of $\mathcal{X}$. Coverage of a Hellman matrix is defined as the number of distinct entries in this matrix, or the number of elements in the set $H = \bigcup_{i=1}^{m} \bigcup_{j=0}^{t-1} f^j(x_{i0})$.

In the paper [1], lower and upper bound for the expected coverage of a Hellman matrix is obtained as follows

$$\sum_{i=1}^{m} \sum_{j=0}^{t-1} \left(1 - \frac{it}{N}\right)^{j+1} \leq \mathbb{E}[|H|] \leq mt. \qquad (1)$$

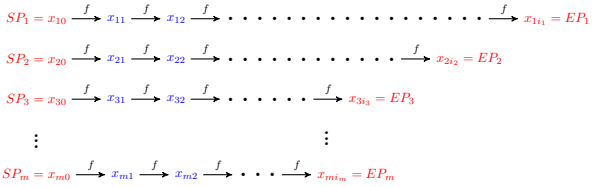
One of the main consequences of overlap situations is the *false alarm* phenomena. Let image point $y \in \mathcal{X}$ is given. There exist situations in which the chain that has been produced from y, in the online phase, merges with some other chain that is saved in the Hellman table, which unfortunately does not include y (see the Figure 1 (b)). Therefore, looking for a matching end point does not essentially imply that the image y will be found in the regenerated chain through the related start point. As illustrated in Figure 1 (b), a wrong starting point $x_{i_1 0}$, instead of $x_{i_2 0}$, will be searched in the online phase and x can't be found.

In order to construct a perfect table version of the Hellman's method, we will consider a generic procedure, introduced in the papers [6, 8], for constructing perfect chains in perecomputation phase. Let $S_f(x) = \{f^t(x) : t \geq 0\}$ be the set of all distinct elements in the chain starting from $x \in \mathcal{X}$. At this point we describe the procedure of generating perfect chains:

- Given a random point $x_{10} \in \mathcal{X}$. We generate the chain starting from x_{10} and we terminate construction of the chain, when a loop appears. The event for the number of points in this chain is denoted by Ch_1, or equivalently $\mathrm{Ch}_1 = |S_f(x_{10})|$.
- For $i \geq 2$, a random point $x_{i0} \in \mathcal{X}$ is chosen and we generate the chain starting from x_{i0}. When the chain merged with a previously constructed chain or when a loop occured, we stop the process. The event for the number of elements in i'th chain is denoted by Ch_i. Note that

$$\mathrm{Ch}_i = \left| S_f(x_{i0}) / \left(\bigcup_{j=1}^{i-1} S_f(x_{j0}) \right) \right| \qquad \text{for } i \geq 2.$$

We follow this construction m times as follows which results in m perfect chains, where $\mathrm{Ch}_j = i_j$ for $j = 1, 2, \ldots, m$.



As you observe, this method ends up with a number of chains with variable lengths, where all elements in chains are distinct.

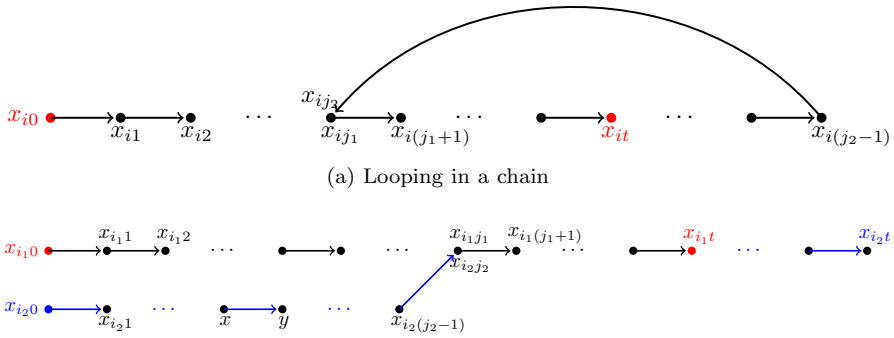
The main disadvantage of this procedure, which make it probably impractical, is that every single element of every chain that is produced, has to be looked up in all formerly generated chains. For this reason, Avoine et al. [6] stated that constructing perfect Hellman tables is not efficient. In [8], the authors suggest to keep the record of points already included in to chains efficiently by using interval trees in the precomputation phase. However these look ups increase the precomputation complexity, the actual profits of the method which relies on perfect chains depends both on the implementation of look ups and characteristics of the chains. On the other hand, in the online phase of this approach there is no false alarm and it is possible to have a high probability of success. By the way, using most efficient cycle detection and collision detection algorithms [9–11] in the precomputation phase, we can decrease the precomputation complexity.

In the next section, we present a probabilistic analysis of the perfect chains characteristics. More precisely, we are interested in the expected values and variances of the random variables Ch_i, and

$$C_i = \mathrm{Ch}_1 + \cdots + \mathrm{Ch}_i = \bigcup_{j=1}^{i} S_f(x_{j,0}), \qquad \text{for } 1 \leq i \leq m.$$

Note that, the random variable Ch_i indicates the coverage of i'th perfect chain and the random variable C_i indicates the coverage i perfect chains.

In [6], the authors provide the probability of success of a single perfect Hellman table and the expected cryptanalysis time. The authors of the paper [8], analyzed the characteristics of perfect chains, such as the probabilities that a chain will be of a certain length, the number of chains, their average lengths, and the coverage of the search space, theoretically and experimentally. Even though perfect Hellman tables have been analyzed in these two references, yet until right now, there is no explicit expression regarding characteristics of the perfect chains, like expected values and variances of the random variables C_i and Ch_i.



(a) Looping in a chain

(b) Merging in two chains

Figure 1. Merging and Looping in Chains

3 Analysis of the Perfect Chains

In this section, we are going to discuss about expected values of the random variables Ch_i and C_i for $1 \le i \le m$. Since we supposed that f is a random mapping, we can assume that we have an urn containing N distinctly marked balls and these balls are drawn from the urn, one at a time, with replacements. Indeed, the random variable Ch_1 counts the number of balls that must be drawn in order to obtain a repeated value. So we have

$$\Pr[\mathrm{Ch}_1 = k_1] = \left(\prod_{i=0}^{k_1-1} \frac{N-i}{N}\right) \frac{k_1}{N}$$

$$\Pr[\mathrm{Ch}_2 = k_2 | \mathrm{Ch}_1 = k_1] = \left(\prod_{i=k_1}^{k_2-1} \frac{N-i}{N}\right) \frac{k_1+k_2}{N}$$

$$\Pr[\mathrm{Ch}_2 = k_2 | \mathrm{Ch}_1 = k_1, \mathrm{Ch}_2 = k_2] = \left(\prod_{i=k_1+k_2}^{k_3-1} \frac{N-i}{N}\right) \frac{k_1+k_2+k_3}{N}$$

$$\vdots$$

$$\Pr[\mathrm{Ch}_m = k_m | \mathrm{Ch}_1 = k_1, \ldots, \mathrm{Ch}_{m-1} = k_{m-1}] = \left(\prod_{i=k_1+\cdots+k_{m-1}}^{k_m-1} \frac{N-i}{N}\right) \frac{k_1+\cdots+k_m}{N}$$

By multiplying above conditional probabilities, we have

$$\Pr[C_m = k] = \sum_{\substack{k_1+\cdots+k_m=k \\ k_1\ge 1, k_2, \ldots, k_m \ge 0}} \Pr[\mathrm{Ch}_1 = k_1, \ldots, \mathrm{Ch}_m = k_m] = \left(\prod_{i=0}^{k-1} \frac{N-i}{N}\right) \frac{k}{N^m} P_{m-1}(k),$$

where,

$$P_{m-1}(k) = \sum_{\substack{k_1+\cdots+k_m=k \\ k_1\ge 1, k_2, \ldots, k_m \ge 0}} k_1(k_1+k_2)\cdots(k_1+\cdots+k_{m-1}).$$

Let $l_1 = k_1, l_2 = k_1+k_2, \ldots, l_{m-1} = k_1+\cdots+k_{m-1}$. Then $1 \le l_1 \le l_2 \le \cdots \le l_{m-1} \le k$ and we obtain that

$$P_{m-1}(k) = \sum_{l_{m-1}=0}^{k} l_{m-1} \sum_{l_{m-2}=0}^{l_{m-1}} l_{m-1}\cdots l_2 \sum_{l_1=0}^{l_2} l_1.$$

So $P_m(k) = \sum_{x=0}^{k} x P_{m-1}(x)$, which satisfy in assumptions of the Lemma 2.

Since as $i \ll N$, $\frac{N-i}{N} \approx e^{-i/N}$ [4, Appendix A], the asymptotic expected value of the random variable C_m, as N tends to infinity, is as follows

$$\begin{aligned}
\mathbb{E}[C_m] &= \sum_{k=0}^{N} k \Pr[C_m = k] \\
&= \frac{1}{N^m} \sum_{k=0}^{N} \left(\prod_{i=0}^{k-1} \frac{N-i}{N}\right) k^2 P_{m-1}(k) \\
&\underset{i\ll N}{\approx} \frac{1}{N^m} \sum_{k=0}^{N} k^2 P_{m-1}(k) e^{-\frac{k^2}{2N}} \\
&\underset{N\to\infty}{\approx} \frac{N}{N^m} \int_0^1 (Nx)^2 P_{m-1}(Nx) e^{-\frac{N}{2}x^2} \mathrm{d}x.
\end{aligned}$$

Let $P_{m-1}(x) = \sum_{j=0}^{2m-2} a_{m-1,j} x^j$. Therefore as $N \to \infty$ we obtain that

$$\begin{aligned}
\mathbb{E}[C_m] &\approx \frac{1}{N^{m-3}} \int_0^1 x^2 \left(\sum_{j=0}^{2m-2} a_{m-1,j} (Nx)^j\right) e^{-\frac{N}{2}x^2} \mathrm{d}x \\
&= \sum_{j=0}^{2m-2} a_{m-1,j} \frac{N^j}{N^{m-3}} \left(\int_0^1 x^{j+2} e^{-\frac{N}{2}x^2} \mathrm{d}x\right).
\end{aligned}$$

So by using Lemmas 2 and 3 we easily conclude that,

$$\mathbb{E}[C_m] \approx \frac{(2m-1)!}{(2^{m-1}(m-1)!)^2} \sqrt{\frac{\pi N}{2}} + O(1). \qquad (2)$$

The Equation (2), indicates the asymptotic ex-

pected value of the coverage of m perfect chains, as N tends to infinity.

Similar to above procedure and by using Lemmas 2 and 3 we derive

$$\begin{aligned}\mathbb{E}[C_m^2] &= \sum_{k=0}^{N} k^2 \Pr[C_m = k] \\ &\approx \sum_{j=0}^{2m-2} a_{m-1,j} \frac{N^j}{N^{m-4}} \left(\int_0^1 x^{j+3} e^{-\frac{N}{2}x^2} \mathrm{d}x \right) \\ &\approx 2mN + O(\sqrt{N}).\end{aligned}$$

Thus, the asymptotic value for the variance of the coverage of m perfect chains is obtained as follows, as N tends to infinity.

$$\begin{aligned}\mathbb{V}[C_m] &= \mathbb{E}[C_m^2] - (\mathbb{E}[C_m])^2 \\ &\approx \left(2m - \frac{\pi}{2} \left(\frac{(2m-1)!}{(2^{m-1}(m-1)!)^2} \right)^2 \right) N + O(\sqrt{N}).\end{aligned} \tag{3}$$

Now, since $\mathbb{E}[\mathrm{Ch}_m] = \mathbb{E}[C_m] - \mathbb{E}[C_{m-1}]$, we easily conclude that, as N tends to infinity

$$\mathbb{E}[\mathrm{Ch}_m] \approx \frac{2m(2m-1)!}{(2^m(m)!)^2} \sqrt{\frac{\pi N}{2}} + O(1). \tag{4}$$

Note that by using Equation (4), $\mathbb{E}[\mathrm{Ch}_1] \approx \sqrt{\frac{\pi N}{2}}$. This result has been investigated previously in the standard birthday problem for a year with N days; for example see [15, 16] and the brilliant paper [13].

Evidently, as $m \to \infty$, by using the Sterling formula, $m! \approx m^m e^{-m} \sqrt{2\pi m}$, and the Equations (2) and (4) we have

$$\mathbb{E}[C_m] \approx \sqrt{2N(m-1)}, \quad \text{as } m \to \infty, \tag{5}$$

$$\mathbb{E}[\mathrm{Ch}_m] \approx \sqrt{\frac{N}{2(m-1)}}, \quad \text{as } m \to \infty. \tag{6}$$

So we obtain that

$$\mathbb{E}[C_m]\mathbb{E}[\mathrm{Ch}_m] \approx N, \quad \text{as } m \to \infty. \tag{7}$$

The Equations (2), (4), (5), (6) and (7) indicates how one can choose parameters to obtain an efficient perfect table. For example, when we want to have a full coverage of the search space, then the average number of the perfect chains is obtained as follows

$$\mathbb{E}[C_m] \approx \sqrt{2N(m-1)} = N \Rightarrow m = \frac{N}{2} + 1.$$

Also, for $m = \frac{N}{2} + 1$, we have $\mathbb{E}[\mathrm{Ch}_m] \approx 1$ is the minimum chain length.

The perfect and non-perfect Hellman's TMTO methods for different values of m, with matrix stopping rule $mt^2 = N$, are compared in the Table 1.

4 Experimental Results

During this section, we provide some experiments which are carried out to check of our theoretic results made in the former section. The numerical results are calculated using the Matlab software.

In our tests, the random mapping f is constructed from the mini-AES (16-bit) cipher [17]. Our reasons for choosing this specific algorithm are: (1) The search space of mini-AES is small against AES algorithm and we can use this algorithm more times for testing our theoretical data; (2) almost all functions derived from mini-AES algorithm, using a constant plaintext, have properties of a random function [13].

More explicitly, the random mapping $f : \mathcal{X} \to \mathcal{X}$ maps the 16-bit key in $\mathcal{X} = \{0,1\}^{16}$ to 16-bit ciphertext in $\mathcal{X}$ under a fixed plaintext. Different fixed plaintexts are used to create multiple random mappings. Also, starting points $x_{i,0} \in \mathcal{X}$, $1 \le i \le m$, are generated randomly. Then we explicitly computed

$$\begin{aligned}&\mathrm{Ch}_1 = |S_f(x_{1,0})|, \quad \mathrm{Ch}_i = |S_f(x_{m,0}) / \cup_{j=1}^{i-1} S_f(x_{j,0})|, \\ &C_1 = |S_f(x_{1,0})|, \quad C_i = \cup_{j=1}^{i} S_f(x_{j,0}),\end{aligned}$$

for $2 \le i \le m$, on the search space of size $N = 2^{16}$. Finally, numerous test instances are conducted for different random starting points. The average values in our test instances are given in Figure 2 ((a) and (b)). It can be seen that the experimental curve is nearly close to the theoretical curve, obtained by the Equations (2) and (4).

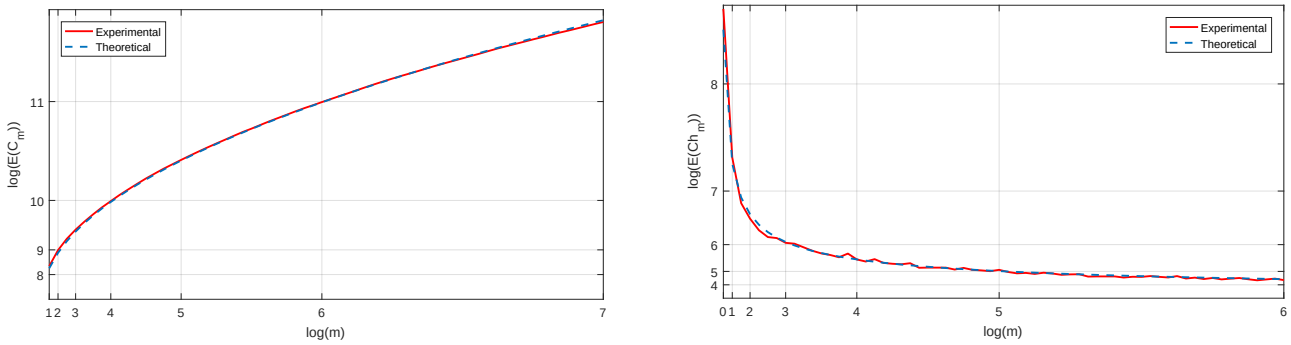
5 Summary and Conclusions

Within the basic Hellman's TMTO approach, points are generated and immediately contained into the current chain until a fixed chain length is achieved, and after that we start the creation of the next chain. In the case of perfect Hellman's TMTO approach, the current chain is finished if a previously employed point happens again. This method results in a number of chains with variable lengths where all points inside all chains are distinct and hence 100% coverage of the search space can be guaranteed. Such perfect chains would certainly ensure the discovering of the solutions of the equation $f(x) = y$, thus, making this technique deterministic.

This paper analyzed the process of the perfect chain behavior in the Hellman's TMTO approach, under an assumption that the searching of points previously covered by earlier chains, using an efficient collision detection algorithm is possible. In fact, we presented the theoretical and practical analysis that determines characteristics of the perfect chains, such as the expected values and variances of the perfect chains. In or-

Table 1. Comparison of the Perfect and Non-Perfect Hellman's TMTO Methods, for Different Values Of m With $mt^2 = N$.

	Hellman's TMTO method		Perfect version		
m	chain length	expected coverage	min chain length	max chain length	coverage
$N^{1/3}$	$N^{1/3}$	$0.6\sqrt{2}N^{2/3}$	$\frac{1}{\sqrt{2}}N^{1/3}$	$\sqrt{\frac{\pi N}{2}}$	$\sqrt{2}N^{2/3}$
$N^{2/3}$	$N^{1/6}$	$0.6\sqrt{2}(N^{5/6} - \frac{1}{2}N^{1/2})$	$\frac{1}{\sqrt{2}}N^{1/6}$	$\sqrt{\frac{\pi N}{2}}$	$\sqrt{2}N^{5/6}$
$\frac{N}{4}$	2	$\frac{1}{2\sqrt{2}}N$	$\sqrt{2}$	$\sqrt{\frac{\pi N}{2}}$	$\frac{1}{\sqrt{2}}N$



(a) $\mathbb{E}[C_m]$ (b) $\mathbb{E}[\mathrm{Ch}_m]$

Figure 2. Theoretically and Experimentally Obtained $\mathbb{E}[C_m]$ And $\mathbb{E}[\mathrm{Ch}_m]$, for $1 \leq m \leq 2^7$ And $N = 2^{16}$ (With mini-AES Algorithm).

der to obtain these characteristics, we used an asymptotic probabilistic analytical model, which treats the chain generation as a random process.

Although, for large search spaces, loop and collision detection algorithms for generating perfect chains in the precomputation phase may not be practically feasible, our analysis provides a deeper understanding of the statistical details of the perfect chains.

References

[1] Martin Hellman. A cryptanalytic time-memory trade-off. *IEEE transactions on Information Theory*, 26(4):401–406, 1980.

[2] Dorothy Elizabeth Robling Denning. *Cryptography and data security*. Addison-Wesley Longman Publishing Co., Inc., 1982.

[3] Philippe Oechslin. Making a Faster Cryptanalytic Time-Memory Trade-Off. In *Crypto*, volume 2729, pages 617–630. Springer, 2003.

[4] Jin Hong and Sunghwan Moon. A comparison of cryptanalytic tradeoff algorithms. *Journal of cryptology*, 26(4):559–637, 2013.

[5] Ga Won Lee and Jin Hong. Comparison of perfect table cryptanalytic tradeoff algorithms. *Designs, Codes and Cryptography*, 80(3):473–523, 2016.

[6] Gildas Avoine, Pascal Junod, and Philippe Oechslin. Characterization and improvement of time-memory trade-off based on perfect tables. *ACM Transactions on Information and System Security (TISSEC)*, 11(4):17, 2008.

[7] Johan Borst, Bart Preneel, and Joos Vandewalle. On the time-memory tradeoff between exhaustive key search and table precomputation. In *Symposium on Information Theory in the Benelux*, pages 111–118. TECHNISCHE UNIVERSITEIT DELFT, 1998.

[8] Violeta Tomašević and Milo Tomašević. An analysis of chain characteristics in the cryptanalytic TMTO method. *Theoretical Computer Science*, 501:52–61, 2013.

[9] Antoine Joux. *Algorithmic cryptanalysis*. CRC Press, 2009.

[10] Gabriel Nivasch. Cycle detection using a stack. *Information Processing Letters*, 90(3):135–140, 2004.

[11] Paul C Van Oorschot and Michael J Wiener. Parallel collision search with cryptanalytic applications. *Journal of cryptology*, 12(1):1–28, 1999.

[12] Jin Hong. Perfect Rainbow Tradeoff with Checkpoints Revisited. *PloS one*, 11(11):e0166404, 2016.

[13] Philippe Flajolet and Andrew M Odlyzko. Random mapping statistics. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 329–354. Springer, 1989.

[14] Valentin Fedorovich Kolchin. *Random graphs*, volume 53. Cambridge University Press, 1999.

[15] Bernard Harris et al. Probability distributions related to random mappings. *The Annals of*

Mathematical Statistics, 31(4):1045–1062, 1960.

[16] Herman Rubin and Rosedith Sitgreaves. Probability distributions related to random transformations of a finite set. Technical report, Stanford Univ Ca Applied Mathematics and Statistics Labs, 1954.

[17] Raphael Chung-Wei Phan. Mini advanced encryption standard (mini-AES): a testbed for cryptanalysis students. *Cryptologia*, 26(4):283–306, 2002.

[18] John H Conway and Richard Guy. *The book of numbers*. Springer Science & Business Media, 2012.

A Some Lemmas

The following lemmas are used in this work.

Lemma 1 (Faulhaber's formula [18]). *Let* $P(x) = \sum_{k=1}^{x} k^p = 1^p + 2^p + \cdots + x^p$. *Then* $P(x)$ *is a polynomial of degree* $p+1$ *with the fillowing form*

$$P(x) = \frac{1}{p+1}\sum_{i=0}^{p}(-1)^i \binom{p+1}{i} B_i x^{p+1-i},$$

where $B_i = \sum_{k=0}^{i}\sum_{v=0}^{k}(-1)^v \binom{v}{k}\frac{v^m}{k+1}$, *for* $i = 0, 1, \ldots$, *are the Bernoulli numbers.*

Lemma 2. *Let* $P_m(x) = \sum_{j=0}^{2m} a_{m,j}x^j$ *be a polynomial of degree* $2m$ *such that*

$$P_m(x) = \sum_{i=m}^{x-1} iP_{m-1}(i), \qquad P_0(x) = 1,\ m \geq 1.$$

Then

$$a_{m,2m} = \frac{1}{2^m m!}.$$

Proof. Let $P_m(x) = \sum_{j=0}^{2m} a_{m,j}x^j$. So $P_{m-1}(x) = \sum_{j=0}^{2m-2} a_{m-1,j}x^j$ and

$$\begin{aligned} P_m(k) &= \sum_{x=m}^{k-1} xP_{m-1}(x) = \sum_{x=m}^{k-1} x\left(\sum_{j=0}^{2m-2} a_{m-1,j}x^j\right) \\ &= \sum_{j=0}^{2m-2} a_{m-1,j}\left(\sum_{x=m}^{k-1} x^{j+1}\right) \\ &= \sum_{j=0}^{2m-2} a_{m-1,j}\left(\sum_{x=0}^{k-1} x^{j+1} - \sum_{x=0}^{m-1} x^{j+1}\right). \end{aligned}$$

Therefore by using the Lemma 1 we have

$$\begin{aligned} P_m(k) &= \sum_{j=0}^{2m-2} a_{m-1,j}\left(\frac{1}{j+2}k^{j+2} + O(k^{j+1})\right) \\ &= \frac{a_{m-1,2m-2}}{2m}k^{2m} + O(k^{2m-1}). \end{aligned}$$

So, we obtain that

$$a_{m,2m} = \frac{a_{m-1,2m-2}}{2m}, \qquad a_{0,0} = 1,\ a_{1,2} = \frac{1}{2}. \tag{A.1}$$

Solving the recursive relation (A.1) is easy and

$$a_{m,2m} = \frac{1}{2\times 4\times\cdots\times 2m} = \frac{1}{2^m m!}. \qquad \square$$

Lemma 3. *Let* $I_j = \int_0^1 e^{-\frac{N}{2}x^2}x^j \mathrm{d}x$ *for* $j \geq 0$. *Then as* $N \to \infty$ *we have the following asymptotic formulas*

$$I_{2j} \approx \frac{(2j)!}{j!2^j N^j}\sqrt{\frac{\pi}{2N}}, \qquad I_{2j+1} \approx \frac{j!2^j}{N^{j+1}}.$$

Proof. First note that

$$I_0 = \int_0^1 e^{-\frac{N}{2}x^2}\mathrm{d}x = \sqrt{\frac{2\pi}{N}}\int_0^{\sqrt{N}} \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}y^2} \underset{N\to\infty}{\approx} \sqrt{\frac{\pi}{2N}}$$

and

$$I_1 = \int_0^1 e^{-\frac{N}{2}x^2}x\mathrm{d}x = \frac{1}{N}\left(1 - e^{-\frac{N}{2}}\right).$$

Let $\mathrm{d}u = e^{-\frac{N}{2}x^2}x\mathrm{d}x$ and $v = x^{j-1}$ for $j \geq 2$. We obtain that

$$I_j = -\frac{1}{N}e^{-\frac{N}{2}} + \frac{j-1}{N}I_{j-2}, \qquad j \geq 2.$$

As N tends to infinity $I_j \approx \frac{j-1}{N}I_{j-2}$ and so by solving this recursive relation we have

$$\begin{aligned} I_{2j} &\approx \frac{1\times 3\times 5\times\cdots\times(2j-1)}{N^j}I_0 \approx \frac{(2j)!}{j!2^j N^j}\sqrt{\frac{\pi}{2N}}, \\ I_{2j+1} &\approx \frac{2\times 4\times 6\times\cdots\times(2j)}{N^{j+1}}I_1 \approx \frac{j!2^j}{N^{j+1}} \end{aligned} \qquad \square$$

Naser Hossein Gharavi received his B.Sc. in Telecommunication Engineering from University of Tehran, M.Sc. degree in Cryptography from Imam Hossein University and Ph.D. degree in Cyber Defense Engineering from Imam Hossein University, Tehran, Iran, in 1996, 2002 and 2017, respectively. His research interests include cryptography, cryptanalysis, security protocols and network security.

Abdolrasoul Mirghadri received his B.Sc., M.Sc. and Ph.D. degrees in Mathematical Statistics, from the Faculty of Science, Shiraz University in 1986, 1989 and 2001, respectively. He is an associate professor at the faculty and research center of communication and information technology, Imam Hussein University, Tehran, Iran since 1989. His research interest includes: Cryptography, Statistics and Stochastic Processes. He is a member of ISC scientific society.

Mohammad Abdollahi Azgomi received his B.Sc., M.Sc. and Ph.D. degrees in computer engineering (software) (1991, 1996 and 2005, respectively) from Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. His research interests include modeling and evaluation of security, privacy and trust, network security, and software security. He is currently an associate professor at School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.

Sayyed Ahmad Mousavi received his B.Sc. degree in Mathematics from University of Kashan, M.Sc. and Ph. D. degrees in Mathematics from Shahid Bahonar University of Kerman. His research interests focus on quantum information science, matrix analysis, and cryptanalysis.