Success Probability of the Hellman Trade-off

Daegun Ma [1] and Jin Hong [1,2,*]

Department of Mathematical Sciences and ISaC-RIM, Seoul National University, Seoul, 151-747, Korea

Abstract

Cryptanalytic time memory trade-off is a probabilistic algorithm for inverting a generic one-way function. Since its first introduction by Hellman, many variants and their analysis results have appeared. We present a new estimate for the success probability of the original Hellman trade-off, that is more accurate than the lower bound that is widely being used today.

Key words: analysis of algorithms, cryptography, time memory trade-off, Hellman table

1. Introduction

Let $f : \mathcal{X} \to \mathcal{Y}$ be any one-way function, i.e., a function which is easy to compute, but which is hard to invert. An example of interest would be the function mapping a secret key to the encryption of a specific fixed known plaintext. A way to efficiently invert this map would imply total breakdown of the encryption system. In fact, much of cryptanalysis can be interpreted as the process of inverting an appropriate one-way function.

There are two trivial ways to invert a generic one-way function. Given a target $y = f(x) \in \mathcal{Y}$ to invert, one may exhaustively search for $x' \in \mathcal{X}$ such that $f(x') = y$. One could also choose to pre-compute and store the pairs $(x, f(x))$ in a table. Then, when a target $y = f(x) \in \mathcal{Y}$ is given, one can search for it among the second components of the table and give the corresponding first component as an answer. Whereas the exhaustive search method takes a long time, the table lookup method requires a large storage space. Cryptanalytic time memory trade-off (TMTO) is a technique that comes between these two extremes. It inverts a one-way function in time shorter than the exhaustive search method, using a storage smaller than the table lookup method.

The first TMTO algorithm was given by Hellman [5], and many of its extensions [1,3] and variants [2,6,8] have appeared. All cryptanalytic TMTO algorithms consist of two phases. For a fixed one-way function to invert, certain tables are created in the pre-computation phase. Then, the inversion of a given target point is done in the online phase, utilizing the pre-computed tables. TMTO is a probabilistic process whose success probability depends on the created tables.

The asymptotic behaviors of various TMTO algorithms are well understood up to small constant factors, and it has even been shown [2] that, in a certain sense, the algorithms we have are already optimal. On the other hand, practical use of TMTO still requires much experience. One has to choose a TMTO algorithm, and for a fair comparison between algorithms, their success probabilities have to be computable. The comparison is further complicated by the choice of table storage techniques affecting storage size, the less understood behavior of the so called

[*] Corresponding author.
Email addresses: madgun7@snu.ac.kr (Daegun Ma), jinhong@snu.ac.kr (Jin Hong).
[1] Partially supported by The Attached Institute of ETRI.
[2] Partially supported by the BK21 project.

false alarms contributing to online time, and the fact that these issues are all interrelated.

Let us briefly discuss a concrete example. The rainbow table method [8] is a widely used trade-off technique. Under typical parameters for the Hellman and the rainbow trade-offs that naturally correspond to each other and lead to equal storage sizes [3], experience shows that the success probability of the former is higher than the latter by about 2 percentage points. When this probability difference is leveled by increasing the rainbow chain lengths, it results in a 15.4% increase of the rainbow online time with no change to storage size. Hence a more accurate evaluation of the success probability will have a meaningful impact on algorithm comparisons.

Any practical approach to the TMTO techniques leads one to consider their success probabilities, and while previous works [7,8] did take success probabilities into account, our knowledge of the success probability for the Hellman trade-off is less than satisfactory, especially when compared to our corresponding knowledge for the rainbow table method.

In this work, we present evidence that the previous known lower bounds on the success probability of the Hellman trade-off are not very tight and give a new approximation for this probability that closely matches experiment data.

2. Coverage rate of the Hellman matrix

Let us explain the central concepts of this work, and point out what needs to be improved.

Hellman matrix We will briefly explain a very small part of the Hellman trade-off algorithm, referring readers to the original paper [5] for the complete algorithm.

We shall fix the search space of size N to

$$\mathcal{X} = \mathcal{Y} = \{0, 1, \dots, N-1\},$$

for notational simplicity. There are positive integer parameters t and m, that are usually chosen to satisfy the *matrix stopping rule*, $mt^2 = N$. As part of the pre-computation, m points from $\mathcal{X}$, which we denote as $X_{1,0}, \dots, X_{m,0}$, are chosen, randomly and each independently of other points. Then, for each fixed $1 \le i \le m$, the points

$$X_{i,j+1} = f(X_{i,j}) \quad (0 \le j < t)$$

are iteratively computed. The resulting $m \times (t+1)$ matrix $H = (X_{i,j})$ is called the Hellman matrix. The first and last columns of H are gathered to form a Hellman table.

Coverage rate We take the $m \times t$ sub-matrix of H, consisting of the first t columns, and collect their entries in a set. That is, we consider

$$\bar{H} = \{X_{i,j} \mid 1 \le i \le m,\ 0 \le j < t\}.$$

As there could be duplications within this set, the set size $|\bar{H}|$ can be strictly smaller than mt. We define the *expected coverage rate* to be

$$\mathrm{ECR}(N, m, t) = E(|\bar{H}|/mt), \tag{1}$$

which is a measure of how efficiently the search space $\mathcal{X}$ has been covered by f-iterations. Here, the average for expectation is taken over all choices of functions f on $\mathcal{X}$ and the starting points $X_{i,0}$, so that the computed value is what is expected of a random function.

Under suitable assumptions concerning certain *reduction functions*, the success probability of the Hellman trade-off can be computed as

$$1 - \Bigl(1 - \mathrm{ECR} \cdot \frac{mt}{N}\Bigr)^{\ell},$$

when ℓ Hellman tables are utilized. Thus to obtain the success probability of the Hellman trade-off, it suffices to understand and evaluate the expected coverage rate of a Hellman matrix. This is the main subject of the current work.

Assuming f to be a random function, Hellman provided the lower bound

$$\mathrm{ECR}(N, m, t) \ge \frac{1}{mt} \sum_{i=1}^{m} \sum_{j=0}^{t-1} \bigl((N - it)/N\bigr)^{j+1}, \tag{2}$$

and states that this numerically evaluates to 0.80, when $mt^2 = N$. The work [7] carefully evaluates the right hand side of the above inequality and provides the bound

$$\mathrm{ECR} \ge \int_0^1 \frac{1 - e^{-x}}{x}\, dx \approx 0.796599, \tag{3}$$

which is valid when $m \gg 1$, $t \gg 1$, and $mt^2 = N$.

Experiments Even though the validity of these bounds were not at question, it was not clear as to how tight these bounds were, so we conducted tests to see the actual coverage rates.

Our one-way function f was constructed from AES-128. More explicitly, the key to ciphertext mapping under a fixed plaintext was used. Since it is not

[3] A non-perfect rainbow table is considered and, in this discussion, we are disregarding implementation specific memory issues addressed in [2] and false alarms.

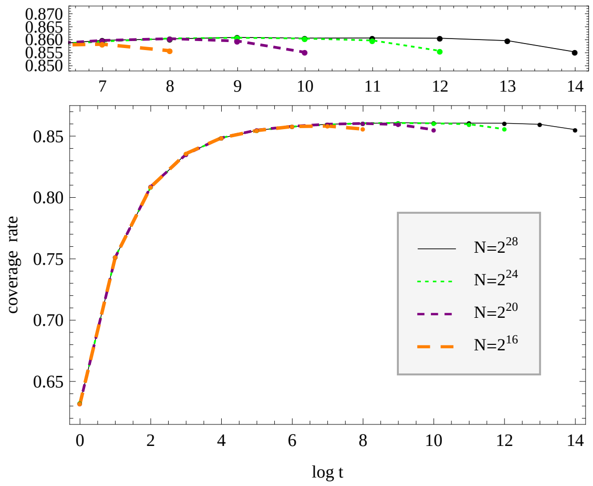

Fig. 1. Real-world coverage rates for $1 \leq t \leq \sqrt{N}$ and its magnified partial view

possible to deal with $N = 2^{128}$, we reduced the input space by fixing most key bits to zero and also truncated the ciphertext accordingly. Even though this choice of f would be widely acceptable, as a precaution, we verified that f exhibits characteristics of a random function [4], by measuring iterated image space sizes. The random starting points $X_{i,0}$ were similarly generated from AES-128 with another fixed plaintext and a counter as input key.

Then we explicitly computed $|\bar{H}|/mt$ on search space sizes N of 2^{16}, 2^{20}, 2^{24}, and 2^{28}, with varying m and t, subject to $mt^2 = N$. Numerous test instances were conducted for each parameter set, with each instance using newly chosen plaintexts. The average coverage rates obtained are given in Fig.1.

It can be seen that the curves for different N are almost indistinguishable from each other for most values of t. The curves part from each other only when one of them nears its maximum possible t of $\sqrt{N}$. For larger N and values of t that would be of interest, it is clear that the actual coverage rate will be much higher than what is given by the current known bound of ECR ≥ 0.80. For example, when $N = 2^{24}$ and $m = t = 2^8$, the observed coverage rate was 0.8605.

3. New estimate for Hellman coverage rate

In this section, we shall give a formula estimating the expected coverage rate of a Hellman matrix more accurately than the previous bounds (2) and (3).

We start with an easy lemma. Let us say we have an urn containing N distinctly marked balls. Suppose m balls are drawn from this urn, one at a time, with replacements. We will compute how many different balls one can expect to see at the end.

Let n_i denote the number of distinct balls one is expected to see up to the i-th draw. For example, one starts with $n_0 = 0$ and $n_1 = 1$. After n_i distinct balls have made their appearance, the probability that the $(i+1)$-th draw will reveal a new ball is $1 - n_i/N$. Hence we have

$$n_{i+1} = n_i + \big(1 - \frac{n_i}{N}\big).$$

Solving this, with the initial condition $n_1 = 1$, gives

$$n_m = N\{1 - \big(1 - \frac{1}{N}\big)^m\}.$$

Recalling that $\exp(-1) = \lim_{N\to\infty}(1 - 1/N)^N$, we arrive at the following lemma.

> **Lemma** When N is large, we can expect to see approximately $N\big(1 - \exp(-m/N)\big)$ distinct balls, when m balls are randomly drawn, with replacements, from an urn containing N balls.

The Hellman bound for coverage rate, given by (2), was obtained by carefully estimating the number of new entries added by each *row* of the Hellman matrix. We shall present a new estimate for coverage rate by going through each *column*, applying the above lemma. This is similar in spirit to the arguments of [8].

For each $0 \leq k < t$, let m_k denote the expectation for the number of distinct entries appearing in the k-th column of the Hellman matrix, which had not appeared in any of the previous columns. Formally, we are setting $\bar{H}_k = \{X_{i,j} \mid 1 \leq i \leq m, 0 \leq j \leq k\}$ and $m_k = E(|\bar{H}_k \setminus \bar{H}_{k-1}|)$, where the average for expectation is taken over all choices of f and $X_{i,0}$. We shall also use the notation $p_k = m_k/N$, so that solving for p_k is equivalent to solving for m_k.

Our Lemma shows that, in the 0-th column, we can expect to find $N(1 - \exp(-m/N))$ distinct entries. So we can start with

$$p_0 = 1 - \exp\big(-\frac{m}{N}\big).$$

Let us find expressions for other m_k terms.

Note that if $X_{i,j} \in \bar{H}_{k-1}$, then $f(X_{i,j}) \in \bar{H}_k$ so that, of the entries $X_{i,k}$ belonging to the k-th column of H, only those belonging to $\bar{H}_k \setminus \bar{H}_{k-1}$ may produce new Hellman matrix entries in the $(k+1)$-th column. Thus, with f modeled as a random function, producing the $(k+1)$-th column can be seen as selecting m_k-many balls, with replacements, from an urn of N balls. Our Lemma tell us that there are $N(1-\exp(-m_k/N))$ distinct entries produced in

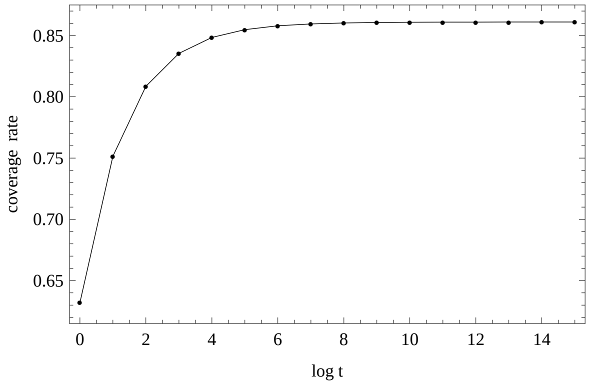

Fig. 2. Theoretically obtained ECR(t) for $1 \le t \le 2^{15}$

the $(k+1)$-th column, which are not automatically old. A certain fraction of these will have appeared before, and we can write

$$m_{k+1} = \big(1 - \sum_{j=0}^{k} p_j\big) \cdot N\big(1 - \exp\big(-\frac{m_k}{N}\big)\big),$$

or equivalently,

$$p_{k+1} = \big(1 - \sum_{j=0}^{k} p_j\big)\big(1 - \exp(-p_k)\big). \qquad (4)$$

Now, using the notation $s_k = \sum_{j=0}^{k-1} p_j$ and the condition $s_1 = p_0 = 1 - \exp(-m/N)$, after some computation, one can turn the above equation into

$$s_0 = 0, \quad s_{k+1} = 1 - \exp\big(-\frac{m}{N}\big)\exp(-s_k). \qquad (5)$$

Finally, by definition of s_k, the expected coverage rate (1) of a Hellman table can now be written as

$$\mathrm{ECR}(N, m, t) = \frac{N}{mt} s_t, \qquad (6)$$

for all $t \ge 1$. Notice that, under the matrix stopping rule $mt^2 = N$, the above is equivalent to

$$s_0 = 0, \quad s_{k+1} = 1 - \exp\big(-\frac{1}{t^2}\big)\exp(-s_k),$$
$$\mathrm{ECR}(t) = t \cdot s_t,$$

which is a function of only t, and not of N or m. The graph of this function is given in Fig.2. Our new estimates are very close to the test results. In fact, except for the $t = \sqrt{N}$ case of each N, most of our theoretic estimates and test results agree up to the third significant digit.

In practice, the starting points $X_{i,0}$ are chosen to be distinct rather than taken independently of each other. In such a case, (4) remains valid with the initial condition $p'_0 = \frac{m}{N}$, and (5) becomes

$$s'_0 = 0, \quad s'_{k+1} = 1 - \big(1 - \frac{m}{N}\big)\exp(-s'_k).$$

We have also verified this claim with experiments.

Finally, we give a closed form approximation for (5). With the Euler method[4] we can derive

$$\frac{d}{dk} s_k = (1 - s_k) - \exp\big(-s_k - \frac{m}{N}\big).$$

Applying $\exp(x) \approx 1 + x + \frac{x^2}{2}$ to this and solving with initial condition $s_0 = 0$ brings one to

$$s_k \approx 2\big(1 - \frac{1}{\tau^2}\big)\frac{e^{k/\tau} - e^{-k/\tau}}{(\tau+1)e^{k/\tau} + (\tau-1)e^{-k/\tau}},$$

where $\tau = \sqrt{\frac{2N}{m}}$.

4. Coverage rate for the $m = 1$ case

The only discrepancy between Fig.1 (experiment) and Fig.2 (theory) is that, with the experiment data, the coverage rates drop slightly as t reaches its maximum possible value of $\sqrt{N}$, or equivalently, when $m = 1$. Explicitly, for $N = 2^{24}$ and $t = 2^{12}$, our tests show ECR $\approx 0.8557 \pm 0.0005$ at 95% confidence, while our theoretically computed ECR is 0.8610 for the same N and t. So we cannot dismiss this difference as experimental error. While the $m = 1$ case is not of practical interest, let us look into this for the sake of completing our understanding of the Hellman trade-off success probability.

Given any $0 < k < t$, for a random walk to intersect itself at length k, its first k elements must be distinct and its $(k+1)$-th element must be one of the previous k elements. Probability of such an event happening is

$$(1)\big(1 - \frac{1}{N}\big)\big(1 - \frac{2}{N}\big)\cdots\big(1 - \frac{k-1}{N}\big) \cdot \frac{k}{N}.$$

Similarly, the probability of reaching the full chain length t without collision is

$$(1)\big(1 - \frac{1}{N}\big)\big(1 - \frac{2}{N}\big)\cdots\big(1 - \frac{t-1}{N}\big).$$

Thus, the exact value of ECR($N, 1, t$) is

$$\frac{1}{t}\Big\{\Big(\sum_{k=1}^{t-1} k \cdot \frac{k}{N} \cdot \prod_{i=0}^{k-1}\big(1 - \frac{i}{N}\big)\Big) + t \cdot \prod_{i=0}^{t-1}\big(1 - \frac{i}{N}\big)\Big\}.$$

Taking note of the identity

$$\tfrac{k}{N}\prod_{i=0}^{k-1}\big(1 - \tfrac{i}{N}\big) = \prod_{i=0}^{k-1}\big(1 - \tfrac{i}{N}\big) - \prod_{i=0}^{k}\big(1 - \tfrac{i}{N}\big),$$

[4] This was suggested by Nurdan Saran during private communication.

one can simplify the above into

$$\mathrm{ECR}(N,1,t) = \frac{1}{t}\sum_{k=1}^{t}\prod_{i=0}^{k-1}(1-\frac{i}{N}). \qquad (7)$$

Numerical computation shows that the above ECR is approximately 0.8556 for $N = 2^{24}$ and $t = 2^{12}$, which is in good agreement with our experiment result of 0.8557.

We can gain more insight into the $m = 1$ case through an approximation of the above ECR. When $i \ll N$, we have $1 - \frac{i}{N} \approx \exp(-\frac{i}{N})$, so that

$$\prod_{i=0}^{k-1}(1-\frac{i}{N}) \approx \exp(-\frac{k^2}{2N}),$$

for $k \ll N$. We can now interpret the sum (7) as an integral and write

$$\begin{aligned}\lim_{t\to\infty}\mathrm{ECR}(N,1,t) &\approx \int_0^1 \exp(-\frac{t^2}{2N}x^2)\,dx\\ &= \sqrt{2\pi}\frac{\sqrt{N}}{t}\int_0^{\frac{t}{\sqrt{N}}}\varphi_{0,1}(z)\,dz,\end{aligned}$$

where $\varphi_{0,1}(z) = \frac{1}{\sqrt{2\pi}}\exp(-z^2/2)$ is the probability density function for the standard normal distribution. Returning to the condition $t = \sqrt{N}$, we can conclude

$$\lim_{N\to\infty}\mathrm{ECR}(N,1,\sqrt{N}) \approx \sqrt{2\pi}\cdot 0.3413 \approx 0.8555.$$

It only remains to explain where we introduce error, when arguments of the previous section are followed with the $m = 1$ case. Let us use the temporary notation $q_k = \prod_{i=0}^{k}(1 - i/N)$. It is easy to see that q_k is the probability for a random chain to be of length strictly greater than k. When $m = 1$, the number of new entries at the k-th column will be either 0 or 1. As this is 1 if and only if a chain is longer than k, the expectation m_k for this number is q_k, and $p_k = m_k/N$ should be equal to q_k/N. In fact, this argument may be seen as another proof of (7), as its right hand side is a sum of the q_k divided by t.

Note that, by definition of q_k, we have

$$\frac{q_0}{N} = \frac{1}{N}, \quad \frac{q_{k+1}}{N} = (1-\frac{k+1}{N})\frac{q_k}{N}. \qquad (8)$$

In comparison, when p_k is small, so that $p_k \approx 1 - \exp(-p_k)$, we can read the previous argument (4) as

$$p_0 = \frac{1}{N}, \quad p_{k+1} \approx (1-\sum_{j=0}^{k}p_j)\cdot p_k. \qquad (9)$$

Since $\sum_{j=0}^{k} p_j \leq \frac{k+1}{N}$, the ECR obtained through (9) will be greater than that obtained through (8). This error will build up with each iteration and explains the discrepancy between the two estimates for the $m = 1$ case.

5. Conclusion

Cryptanalytic time memory trade-offs are probabilistic algorithms for inverting a one-way function. Hellman gave a lower bound for the success probability of his original algorithm, and all further analyses of the Hellman trade-off were done assuming this lower bound to be a good measure of its success probability.

In this work, we first identified a long overlooked gap between the Hellman's lower bound and the actual success probability. Then a new estimate for the coverage rate of a Hellman matrix was given, from which the success probability can directly be computed. We have also verified our theoretic analysis with test results.

As the success probability of a trade-off algorithm is of fundamental importance in any comparison between trade-off algorithms or in their practical use, our analysis sets a more fair and robust working ground for these tasks.

Another contribution of this work is in quantifying the increase in search space coverage induced by the addition of each new column. This gives more insight into the inner workings of the Hellman trade-off and may be of use in future studies.

References

[1] 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), Vol. 11, Issue 4 (July 2008), Article No. 17.

[2] Elad Barkan, Eli Biham, and Adi Shamir, Rigorous bounds on cryptanalytic time/memory tradeoffs. Advances in Cryptology, Proceedings of Crypto 2006, LNCS 4117, Springer-Verlag, pp.1–21, 2006.

[3] Dorothy E. Denning. Cryptography and Data Security (p.100, Ron Rivest's distinguished points observation). Addison-Wesley, 1982.

[4] Philippe Flajolet and Andrew M. Odlyzko, Random mapping statistics. Advances in Cryptology, Proceedings of Eurocrypt '89, LNCS 434, Springer-Verlag, pp.329–354, 1990.

[5] Martin E. Hellman, A cryptanalytic time-memory tradeoff. IEEE Transactions on Information Theory, Vol. IT-26, No.4, pp.401–406, 1980.

[6] Jin Hong, Kyung Chul Jeong, Eun Young Kwon, In-Sok Lee, and Daegun Ma, Variants of the distinguished

point method for cryptanalytic time memory trade-offs. Information Security Practice and Experience, Proceedings of ISPEC 2008, LNCS 4991, Springer-Verlag, pp.131–145, 2008.

[7] Koji Kusuda and Tsutomu Matsumoto, Optimization of time-memory trade-off cryptanalysis and is application to DES, FEAL-32, and Skipjack. IEICE Transactions on Fundamentals, E-79A(1), pp.35–48, 1996.

[8] Philippe Oechslin, Making a faster cryptanalytic time-memory trade-off. Advances in Cryptology, Proceedings of Crypto 2003, LNCS 2729, Springer-Verlag, pp.617–630, 2003.