Estimating How Confidential Encrypted Searches are using Moving Average Bootstrap Method

Alexander A. Towell and Hiroshi Fujinoki
Department of Computer Science
Southern Illinois University Edwardsville
Edwardsville, IL 62026-1656
+1 618 650 3727
{*atowell*, *hfujino*}*@siue.edu*

ABSTRACT

Frequency attacks are a type of attacks adversaries perform to encrypted searches. Adversaries guess the meaning of the encrypted words by observing a large number of encrypted words and map the encrypted words to guessed plain text words using their known histogram. This paper applies an approach of resilience engineering in studying how effective encrypted searches will be for protecting users' privacy when they perform oblivious searches, i.e., searches on their encrypted data. For estimating the number of encrypted words (N*) an adversary needs to observe before the adversary correctly guesses a certain percentage of the observed words with a certain confidence, we developed and evaluated Moving Average Bootstrap (MAB) method. In the experiments we performed, we observed that MAB method reliably accurately estimated N* in 5% of the time a defender would have to wait before the defender could calculate the estimator (N*).

Categories and Subject Descriptors
H.3.3 [**Information Systems**]: Information Search and Retrieval - Information filtering; Relevance feedback; Search process;
E.2 [**Data**]: Data Storage Presentations - Object representation;

Keywords
Encrypted searches, known-plaintext attacks, oblivious searches, bootstrap method, information retrieval, information security, resilience engineering

1. Introduction

As electric information systems have been our essential infrastructure in our society, the risk of unauthorized information leaks has been rising. Such information leaks have become a serious issue since recent popular adoptions of cloud computing could end up with social turmoil. For example, malicious activities conducted by the people we most trust; system administrators at cloud providers; have been threats that may prevent further adoptions, or even the death of, the network-based electronic information systems, if we do not take proper actions against such crimes [1, 2, 3].

The essential challenge here is a trade-off problem between the privacy and the usability in the data we deposit to remote systems, such as cloud servers. To resolve this trade-off problem, researchers invented a concept, called "encrypted search".

A term, "encrypted search", means a concept that lets authorized users to investigate presence of specific words or phrases in a target data set, such as an encrypted document or a database table where stored data is encrypted [4, 5, 6, 7, 8], while the contents, especially the meaning of the data, are hidden from any unauthorized personnel, including the system administrators of a cloud server.

Encrypted search is expected to bring us tremendous benefits in information security. For example, encrypted search will enable us to perform *oblivious searches* in remote database systems. Users would be able to perform specific searches in remote database systems or search engines, such as Google, without anyone, including their system administrators who usually have full access to all the queries issued by their users, knowing what words or phrases are searched.

Despite its potentials, no encrypted search scheme theoretically guarantees perfect confidentiality. There are many ways information may be leaked. For example, encrypted search queries use a substitution cipher where plain text words are substituted with encrypted counterparts in queries. This causes a major vulnerability in encrypted searches.

A major known threat to encrypted search is frequency analysis (aka, "spectral analysis") attacks [9]. If adversaries can estimate the relative frequency of plain text words, then the most frequently occurring encrypted word probably maps to the most frequently occurring plain text word, which is known as "known-plaintext attacks". Adversaries look for a mapping that maximizes the likelihood of seeing those encrypted words under the estimated distribution.

Adversaries may know that an encrypted document is for a specific purpose and, therefore, they may predict some terms or phrases to appear quite often in the document. According to Piantadosi, the text frequency distribution of words in most documents follows Zipf distribution [10]. Adversaries utilize the expected frequency for guessing their plain text counterparts. These systems are at a risk of information leaks.

We applied the bootstrap method in quantifying the level of accuracy adversaries can achieve, while we face two major uncertainty factors: ① as we describe in Section 3, since the actual accuracy in adversary's will fluctuate in unpredictable ways, it is not easy for the defender side to determine the level of accuracy adversaries may have achieved for each particular system, and ② observing a large number of samples (i.e., encrypted words) for accurately estimating N^* takes time for defenders.

To cope with the issues described above, we developed and evaluated a new method that accurately and efficiently estimates the vulnerability by adversaries (how many encrypted words an adversary needs to observe to correctly map a certain number of encrypted words to those in plain text and how likely an adversary can achieve the goal).

The rest of this paper is organized as follows. In Section 2, we review the existing effort to encrypted searches. There has not been much work that quantitatively analyzes the conditions for information leaks by frequency attacks, such as the number of encrypted words an adversary needs to observe for a certain accuracy and how likely it happens. In Section 3, we introduce our moving average bootstrap (MAB) method to accurately and efficiently estimate the achievable accuracy for frequency attacks performed by adversaries. In the attacks, adversaries' accuracy often fluctuates and adversaries may observe potentially a large number of encrypted words, which is theoretically infinity especially if adversaries are system administrators. In Section 4, we present our performance evaluations on the MAB method. Section 5 summarizes our contributions and planned future work, followed by the selected references.

2. Existing Work

Boneh proposed a method to let a third party to perform searches over encrypted e-mail messages, called public-key encryption with keyword search (PEKS) [4]. Boneh designed PEKS in such a way that e-mail messages are encrypted by the public key of an e-mail receiver, while a third party, such as an e-mail server, to perform search for a particular word (e.g., "urgent") in each encrypted message without all the raw contents in the encrypted e-mail exposed to the third party. The core of this method is trapdoors, which are a hash value of a given word in e-mails. Each e-mail receiver creates trapdoors, one for each target word and trapdoors are included in each encrypted e-mail message for searches on the encrypted e-mail messages.

Li extended this concept to allow third parties to perform encrypted searches that allows deviations from exact matching to target encrypted words by enumerating multiple trapdoors, one for each expected deviation [5]. Cao [6], Sun [7], and Kamara [8] proposed to apply encrypted search to enhancing security in cloud computing.

Despite the potentials in the encrypted search schemes, risk of information leaks through guessing the searched words has been identified [11, 12, 13]. Byun [11], Yau [12], and Jeong [13] technically demonstrated that anyone who has access to encrypted data possibly map them to their plain text counterparts.

Use of secure communication channels (e.g., SSL) will be effective in hiding the trapdoors in queries submitted by legitimate users from external adversaries, but use of secure communication channels still can not prevent frequency attacks from internal adversaries, such as malicious administrators, assuming that they can intercept trapdoors within a local host computer, by installing illegal capturing tool or by tampering executables.

Despite the threat from frequency attacks, there has not been much work that delves into quantified analyses on the conditions for when such information leaks exceed a tolerable risk level under various conditions. Rivain proposed a multivariate Gaussian random variable method to estimate the success rate in discovering secret keys under side-channel attacks [14]. Thillard proposed use of "confidence" for evaluating the effectiveness in side-channel attacks [15]. Rivain and Thillard's both proposed a solution against correlation attacks, but not against frequency attacks. Correlation attacks are different from frequency attacks in that adversaries discover the encryption keys to deduce the plain texts in the former, while the latter induces the plain texts directly from the observed encrypted words without discovering their encryption keys.

3. Moving Average Bootstrap (MAB) Method

We applied the bootstrap method to efficiently estimate how vulnerable encrypted searches will be against frequency attacks especially when adversaries are expected to observe a large number of encrypted words for long period of time. Bootstrap is a method for statistical estimation of various properties on population, such as variance, by calculating the properties using only the limited number of samples that follow a distributed approximation [16].

We applied the bootstrap method for estimating the vulnerabilities to encrypted searches from adversaries under the following four assumptions.

(i) Adversaries know, in advance, that an encrypted document is for a specific purpose and, therefore, they predict histograms of the terms or phrases (thus, their objective is mainly to discover who searched for what terms).

(ii) Adversaries have access to every encrypted search query submitted by legitimate users.

(iii) The encrypted search keys are encrypted using word cypher instead of block cypher to allow search engines to perform uni-gram searches for each search key. This class of attacks is known as cipher text-only attack (COA).

(iv) Adversaries are stand-alone. They do not coordinate their frequency attacks. Thus, each of them will see different sequence of samples (i.e., encrypted words) and they do not share their histograms of their samples with other adversaries. This assumption is for simplifying our analyses on vulnerability on encrypted search, to be relaxed in our future work.

To quantify the vulnerability to frequency attacks, we defined a term, "adversary's accuracy". Adversary accuracy (p^*) is the ratio of the number of the correctly mapped text search keys to that of the total encrypted search keys observed by an adversary, which is defined as:

$$\text{adversary accuracy } (p^*) = \delta/n \qquad (1)$$

where:

δ = the number of the correctly mapped text search keys by an adversary

n = the number of encrypted search keys observed by an adversary in a session of his frequency attack

For example, for the first sequence of encrypted words (the words in the following examples are shown as non-encrypted words to make its meaning clear to the audience) in Figure 1 (i), n = 8 (a duplicated word, "world" is counted twice).

(i) "hello world this is Alex and wonderful world"

(ii) "hello Illinois this is Alex and I am fine"

Figure 1: two examples of sequences of words

Assuming that the following words in the first sequence (i) are correctly mapped to their plain words by an adversary: "hello", "world", "Alex" and "wonderful", δ = 4 and p^* = 0.5 (= 4/8). If another sequence of encrypted words (Figure 1 (ii)) follows the previous sequence (i), and if the following encrypted words are correctly mapped to their plain texts by the same adversary: "Illinois", "hello", and "Alex", n = 17 and δ = 7, resulting in $p^* \cong 0.412$ ($\cong$ 7/17).

The primary challenge in accurately estimating the achieved accuracy is fluctuations in adversary accuracy. The inherent fluctuations in adversary accuracy will make accurately estimating the achievable accuracy difficult, which will prohibit defenders from designing information systems with an expected level of resilience against frequency attacks on encrypted searches. Thus, we need an efficient measure of accuracy that is unaffected by such fluctuations.

Answering the above questions is important especially for defenders, since having a systematic method to accurately and efficiently estimate the achievable accuracy by adversaries in given conditions would allow them to design a system that has a specific level of resilience against frequency attacks on encrypted searches. Such a method will provide a foundation for the safety of our information systems in asking security-related critical questions, such as, "would you be ok with someone only being able to read 70% of your private journal with a 1% chance?"

To cope with the expected fluctuations in adversary accuracy, we developed a new estimation method called moving average bootstrap (MAB) method. The proposed estimation method performs smoothing out of fluctuating adversary accuracy using the bootstrap method. The new method smooths out the fluctuation curve as the measure for the adversary's current accuracy using a moving average.

The primary concern in the proposed estimation method is to answer the following question: when is the risk too high that the adversary has achieved a certain level of accuracy? To quantitatively find an answer for the question, we introduced a metric, the estimator for the minimum number of encrypted words an adversary needs to achieve a given accuracy level of p^*, which is represented by N^*.

The concept of "the estimator for the minimum number of encrypted words an adversary needs to observe for achieving a given accuracy level of p^*" for a *specific* attempt of frequency attack, which we represent by "n^*" (instead of "N^*") is visualized in Figure 2. Figure 2 shows the plots of adversary accuracy estimated by an adversary. Mainly due to the randomness in each sequence of encrypted words an adversary observes, the adversary's accuracy will fluctuate, causing some spikes in both up and down ward (we marked one of such upward spikes as "fluke high point" in Figure 2), making the adversary unsure if he has achieved a target accuracy level (e.g., $p^* = 0.36$ was applied to this example). However, if the adversary's accuracy never goes below $p^* = 0.36$ after a certain number of the observed encrypted words, the adversary is sure that he has achieved the accuracy of 36%. We used a symbol, n^*, to represent the number of the encrypted words.

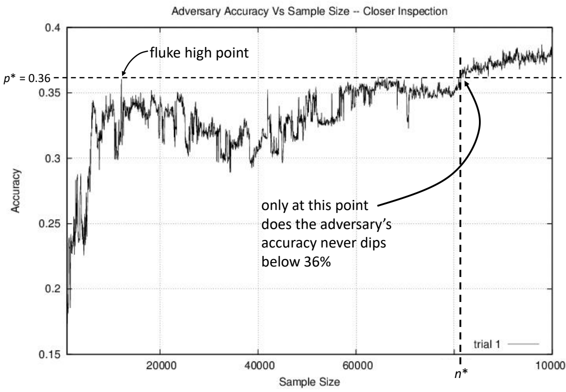

Figure 2: an example of plots of adversary accuracy (p^*) for a session of frequency attack and fluctuations in the adversary accuracy

It is expected that n^* value will be different for each attempt of frequency attack. If three different adversaries perform their own frequency attack to the same target data set, their n^* values for the same accuracy level will be different even if they observed the same number of encrypted words for the same data set just because the sequences of the encrypted words observed in each attempt will be different. Thus, one particular value of n^* will not be reliable enough. As a result, we need to efficiently calculate "the estimator for the minimum number of encrypted search queries an adversary needs to achieve a given accuracy level of p^*" with a reasonably high level of confidence, which is denoted by "N^*". To accurately and efficiently estimate N^* from a limited number of observed n^*'s, while observed adversary accuracies fluctuate in unpredictable manners, we applied the bootstrap method.

We applied the bootstrap method to calculating N^* when adversaries can logically perform a large number of frequency attack sessions, each of which lets each adversary to observe a large number of encrypted words. To achieve the goals, we designed our bootstrap method using the following parameters:

- n (the number of the encrypted words an adversary observes): legitimate users are expected to send search keys as encrypted words so that the search engine can look for matching words over the encrypted words stored in a target data set. The n encrypted words each adversary observes are assumed to be different for each adversary.
- p^* (a level of accuracy an adversary attempts to achieve ($0.0 \le p^* \le 1.0$)): when $p^* = 0.0$, an adversary does not successfully map any encrypted word to its correct word in plain text, while $p^* = 1.0$ means that an adversary successfully mapped all the encrypted words he observed to the correct words (thus, $p^* = 0.5$ means the half of the observed encrypted words are mapped to the correct words in plain text).
- n^* (the minimum number of encrypted search queries an adversary needs to observe to achieve a given accuracy level of p^*): it indicates the smallest value of n in such the way that an adversary never goes below the given accuracy level in an attempt of frequency attack (see Figure 2 for its visualization). Since it is impossible for testing a frequency attack on an infinitely large number of n, we assumed that 10^9 is "infinity" if there was no increase in p^* from $n = 10^9$ to $n = 5 \times 10^9$.
- θ (the probability an adversary achieves an given accuracy level of p^* ($0.0 \le \theta \le 1.0$)): when $\theta = 0.5$, 50% of adversaries, each of whom performed an attempt of frequency attack to the same target data set, will achieve the given accuracy (i.e., successfully mapped ($p^* \cdot 100$)% of his observed encrypted words to their plain counterparts) by observing N^* samples (i.e., encrypted words). $\theta = 1.0$ means every adversary achieves the given accuracy level, while $\theta = 0.0$ means no adversary achieves the given accuracy level.
- k (conjectured sample size for population set): it is the number of frequency attack sessions (each of which produced its own n^*) as our best estimation for an infinitely large number of frequency attack sessions. We justified our selection of k ($k = 5{,}000$) as follows. As we showed in Figure 6 later in this paper, the 95% confidence interval for the population set was [6300, 7550] for $p^* = 0.30$ and $\theta = 0.50$. We repeated the experiments (5,000 samples for our approximation of the population set) 10,000 times, resulting in a total of 50 million samples to produce the 95% confidence interval of 1,250 (i.e., 7,550 – 6,300). The interval is relatively small for the 50 million samples, where their ratio is 0.000025 (i.e., 1,250/50,000,000).
- m (the bootstrap sample size): it is the number of samples (i.e., n^*'s) randomly selected from the set

of k n^*'s ($m << k$). The average of m n^*'s was calculated for a set of m n^*'s.

- s (the number of the bootstrap rounds): it represents how many times calculating the average of m n^*'s is repeated for a given value of p^*. We first randomly selected m n^*'s. Then, we calculated the average of the m n^*'s. Finally, we repeated this s times to calculate N^*.

Using p^*, k, m, s, and θ, the following is the procedure of our method to estimate N^* using the bootstrap method. We first randomly generated k sessions of frequency attacks, which produced k n^*'s (i.e., each session of frequency attack generated a value of n^*). Then, we randomly selected m n^*'s from the set of k n^*'s. From the m n^*'s, we randomly selected m n^*'s, by repeating the following procedure m times:

① selected one n^*'s from the pool of m n^*'s

② put the selected n^* back to the pool of k n^*'s

When a set of m n^*'s were selected, we calculated the average of the m n^*'s. We repeated the above s times, producing s averages of m n^*'s. This way, having only m ($m << \infty$) samples (i.e., m n^*'s), we will be able to efficiently calculate N^* for a large number of frequency attack sessions, improving the accuracy in estimating N^* while it is tractable (estimating N^* from ∞ n^*'s is not tractable). Figure 3 visualizes this concept.

To calculate N^* for a certain confidence level, after we calculated s averages of m n^*'s, by first identifying the bottom and the ceiling for the 95% interval in the following way. We first ordered the s averages of m n^*'s in the ascending order, as $n^*_{[1]}$ through $n^*_{[s]}$. Then, the bottom and the ceiling of a certain percentile were identified so that confidence intervals (we applied 95%) were calculated by throwing out the bottom and top 2.5% of its sampling distribution (Figure 4) as:

$$\text{CI} = mean\{N^*|p^* = 0.36\} \pm 2 \cdot \text{SE}\{N^*|p^* = 0.36\}$$

where "SE" means "Standard Error".

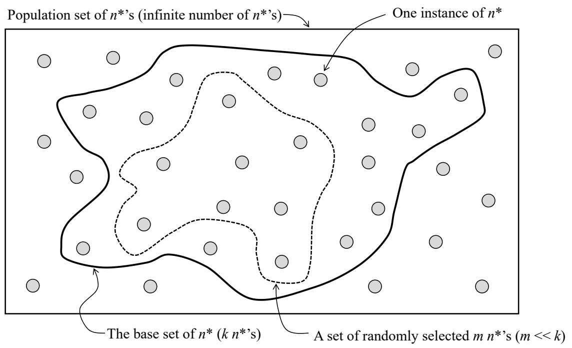
Population set of n*'s (infinite number of n*'s)
One instance of n*
The base set of n* (k n*'s)
A set of randomly selected m n*'s (m << k)

Figure 3: Bootstrapping of m samples

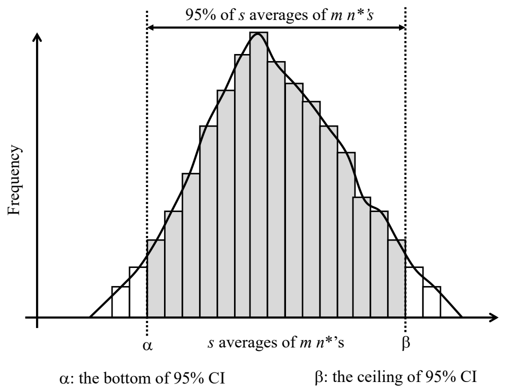

Figure 4: 95% confidence interval for the s averages of m n^*'s

We defined a term "θ quantile" (i.e., "0.05 quantile", if $\theta = 0.05$), to mean "($\theta \cdot 100$) percentile", which is the $n^*_{[(\theta \cdot k)]}$ for the actual distribution and $n^*_{[(\theta \cdot s)]}$ for the empirical distributions. The term, "the actual distribution" means the distribution of the average of k n^*'s, while "the empirical distribution" means the one of the s averages of m n^*'s (using our bootstrap method). We calculated the 95% CI for each percentile we tested.

Finally, we calculated the estimated number of encrypted words that allow an adversary to achieve an adversary accuracy of p^* with a probability of θ (i.e., "N^*") using the following formula:

$$N^* = F^{-1}(\theta \,|\, p^*)$$

4. Performance Evaluation

We compared the performance of our bootstrap method for estimating N^* using only m samples with that of the ideal case (i.e., the one for representing the population set), using a large value k, which will produce N^* close to that for the ∞ sample size. For that purpose, we compared the CDF's of the 5th percentile (i.e., $\theta = 0.05$) for the actual distribution and the empirical distribution.

Figure 5 (a), (b), and (c) show the actual (the population set) and empirical CDF's from our experiments for the following configuration: $p^* = 0.30$, $\theta = 0.05$, $k = 5{,}000$, and $s = 10{,}000$ for $m = 50$, 100, and 250, respectively. The experiments demonstrated the following observations. For a small sampling size ($m = 50$), the actual and empirical sampling distributions are nothing alike. As m (the bootstrap sampling size) increases, the actual and empirical sampling distributions quickly converge to the same shape. At and after $m = 500$, no significant difference was observed between the two distributions.

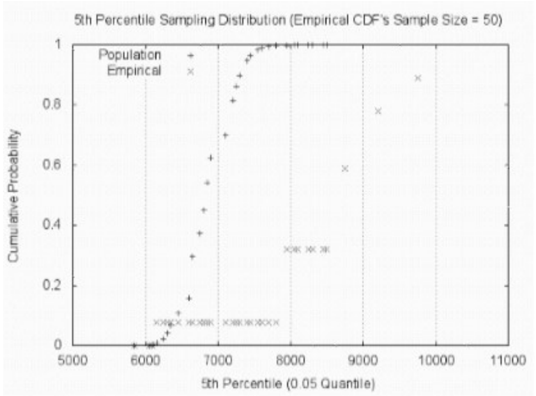

Figure 5 (a): The actual and empirical CDF's of the 5th percentile for *m* (bootstrap sample size) = 50

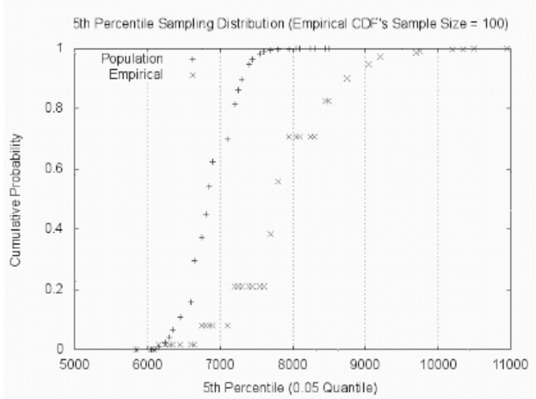

Figure 5 (b): The actual and empirical CDF's of the 5th percentile for *m* = 100

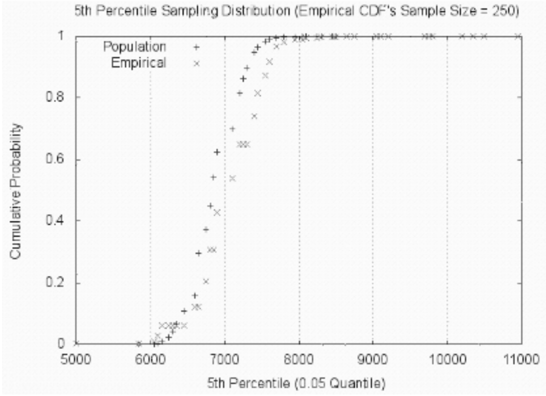

Figure 5 (c): The actual and empirical CDF's of the 5th percentile for *m* = 250

We studied the effect of θ to N^* when we increased θ from 0.01 to 0.50. Our simulation experiments for the above analyses generated approximately 6GB of raw data, from which we made the following observations.

Figure 6 shows the N^*'s, as well as their lower and the upper 95% thresholds, for different levels of the probability (θ = 0.01 through 0.50) an adversary achieves for an accuracy level of p^* = 0.30. The means between the bottom and ceiling of 95% CI (the crosshairs on the solid line) were calculated by taking the means of their sampling distribution.

We repeated the same analyses for the adversary accuracy of 45 and 50% (i.e., p^* = 045 and 0.50). This is a scenario in which an adversary can correctly map 45 and 50% of what the adversary observed by a 5% chance (θ= 0.05), respectively.

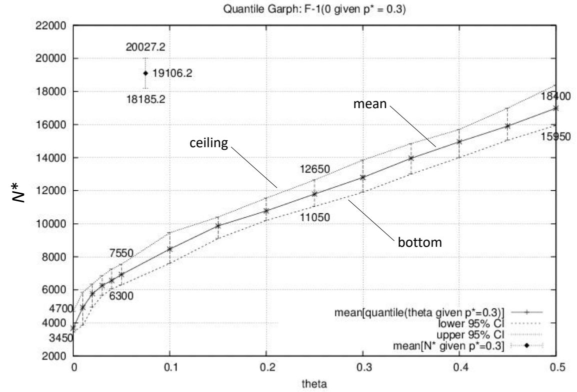

Figure 6: N^* for different levels of the probability for θ = 0.01 through 0.50 an adversary achieves for an adversary accuracy of p^* = 0.30.

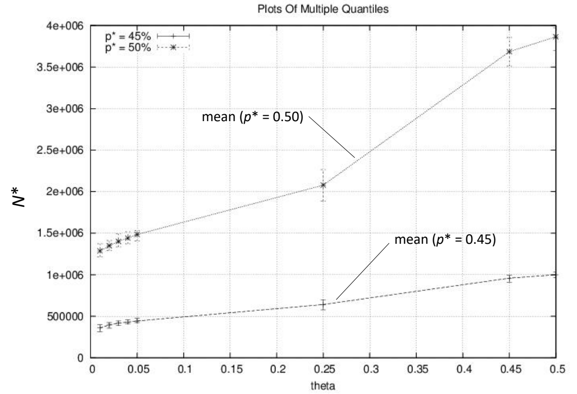

Figure 7: N^* for different levels of the probability for θ = 0.01 through 0.50 an adversary achieves for an accuracy level of p^* = 0.45 and 0.50.

Figure 7 shows the N^*'s to achieve p^* = 0.45 and 0.5 at the probability of 5% (θ= 0.05). We estimated the mean is around 19,100 observations (N^* = 19,100) for p^* = 0.45.

The followings are the observations from our experiments:

- The proposed MAB method calculated the estimated number of encrypted search queries an adversary needs to observe (N^*) for achieving a given accuracy level, p^* = 0.30, at the confidence level of 95% using only 5% of the actual observations (250/5000) (Figure 5 (c)).

- For 5% chance that the adversary can read 30% of what the adversary observed (i.e., $\theta = 0.05$ for $p^* = 0.3$), we estimated that the adversary would need around 6,900 observations (6,900 encrypted words, $N^* = 6{,}900$). The 95% confidence interval of all the s averages of m n^*'s was [α:6,300, β:7,550] encrypted words when $m = 500$ (Figure 6).
- For comparison, for 30% accuracy ($p^* = 0.30$), an adversary only needs around 5,000 encrypted words to be observed ($N^* = 5{,}000$) (Figure 6).
- For a 50% chance of success (i.e., $\theta = 0.50$), we observed that an adversary would need around 17,000 observations ($N^* = 17{,}000$) for 50% of success in achieving $p^* = 0.30$ with $m = 500$ (Figure 7).
- For 45% accuracy (i.e., $p^* = 0.45$), we estimated that an adversary would need a sample size around 360,000 ($N^* = 360{,}000$) to have a 1% chance of success ($\theta = 0.01$) (Figure 7).
- For 50% accuracy (i.e., $p^* = 0.50$), an adversary would need around 1.3 million samples ($N^* = 1.3 \cdot 10^6$) (Figure 7).

5. Conclusions and Future Work

The primary contributions in this paper are two-folds. First, there has not been much work for studying how safe encrypted searches are against frequency attacks, which can be measured by a large number of attackers for long period of time, possibly infinitely long. We provide studies on the resilience of encrypted searches against frequency attacks from the view point of resilience engineering approach to enhance security on encrypted searches. Resilience engineering is a new way of enhancing safety by precisely estimating the level of possible threats to a system and feeding them back to adjusting or re-designing the system to maintain the acceptable level of safety [17].

Our second contribution is development of a new method, Moving Average Bootstrap (MAB) method, which efficiently and accurately calculates the estimator for the minimum number of encrypted words (N^*) an adversary needs to achieve a given accuracy level (p^*) with a certain level of confidence as soon as a relatively small number of samples (n) (i.e., encrypted words) are submitted by legitimate users. Thus, the MAB method will let the defenders calculate the estimator at an early stage without waiting for a large number of queries submitted by legitimate users. Especially from the view point of "tractability", calculating the estimator using, not to mention an infinitely large number of encrypted words, a large number of encrypted words takes time (waiting for a large number of encrypted words to be submitted) and huge storage (storage space to hold the submitted encrypted words) is required.

Our proposed MAB method calculated the estimated number of encrypted search queries an adversary needs to observe (N^*) for achieving a given accuracy level, $p^* = 0.30$, at the confidence level of 95% using only 5% of the actual observations (250/5000) (Figure 5 (c)). Assuming that the increase in the time an adversary needs to achieve a certain p^* is proportional to the ratio in the increase of the number of the encrypted words observed by an adversary (n) for a large number of encrypted words, the MAB method would allow a defender to estimate N^* in 5% of time (without waiting for legitimate users to issue a large number of encrypted words). We are currently performing analyses using higher p^* (0.55 through 0.80) for different levels of confidence (90 to 98%) for observing how they affect the performance of MAB method and for observing if there is any pathological case for MAB method.

Selected References

[1] S. Subashini and V. Kavitha, "A Survey on Security Issues in Service Delivery Models of Cloud Computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1-11, 2011.

[2] Dimitrios Zissis and Dimitrios Lekkas, "Addressing Cloud Computing Security Issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583-592, 2012.

[3] William R. Claycomb and Alex Nicoll, "Insider Threats to Cloud Computing: Directions for New Research Challenges," *Proceedings of IEEE International Conference on Computer Software and Applications*, pp. 388-394, July 2012.

[4] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano, "Public Key Encryption with Keyword Search," *Proceedings of Eurocrypt*, pp. 506-522, May 2004.

[5] Jin Li, Qian Wang, Cong Wang, Ning Cao, Kui Ren, and Wenjing Lou, "Fuzzy Keyword Search over Encrypted Data in Cloud Computing," *Proceedings of IEEE INFOCOM*, pp. 441-445, March 2010.

[6] Ning Cao, Cong Wang, Ming Li, Kui Ren, and Wenjing Lou, "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222-233, January 2014.

[7] Wenhai Sun, Bing Wang, Ning Cao, Ming Li, Wenjing Lou, Thomas Y. Hou, and Hui Li, "Privacy-Preserving Multi-Keyword Text Search in the Cloud Supporting Similarity-Based Ranking," *Proceedings of the ACM Symposium on Information, Computer and Communications Security*, pp. 71-82, May 2013.

[8] Seny Kamara and Kristin Lauter, "Cryptographic Cloud Storage," *Financial Cryptography and Data Security*, *Lecture Notes in Computer Science*, vol. 6054, pp 136-149, 2010.

[9] Jean-François Raymond, "Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems," *Designing Privacy Enhancing Technologies*, *Lecture Notes in Computer Science*, vol. 2009, pp. 10-29, March 2001.

[10] Steven T. Piantadosi, "Zipf's Word Frequency Law in Natural Language: A Critical Review and Future Directions," *Psychonomic Bulletin & Review*, vol. 21, no. 5, pp. 1112-1130, October 2014.

[11] Jin Wook Byun, Hyun Suk Rhee, Hyun-A Park, and Dong Hoon Lee, "Off-Line Keyword Guessing Attacks on Recent Keyword Search Schemes over Encrypted Data," *Secure Data Management*, *Lecture Notes in Computer Science*, vol. 4165, pp. 75-83, 2006.

[12] Wei-Chuen Yau, Swee-Huay Heng, and Bok-Min Goi, "Off-Line Keyword Guessing Attacks on Recent Public Key Encryption with Keyword Search Schemes," Proceedings of the International Conference on Autonomic and Trusted Computing, pp. 100–105, 2008.

[13] Ik Rae Jeong, Jeong Ok Kwon, Dowon Hong, and Dong Hoon Lee. "Constructing PEKS Schemes Secure against Keyword Guessing Attacks is Possible?" *Computer Communications Express*, vol. 32, no. 2, pp. 394-396, 2009.

[14] Matthieu Rivain, "On the Exact Success Rate of Side Channel Analysis in the Gaussian Model," *Selected Areas in Cryptography*, vol. 5381, pp. 165-183, 2009.

[15] Adrian Thillard, Emmanuel Prouff, and Thomas Roche, "Success through Confidence: Evaluating the Effectiveness of a Side-Channel Attack," *Cryptographic Hardware and Embedded Systems*, vol. 8086, pp. 21-36, 2013.

[16] Debashis Kusharya, "Bootstrap Methods and Their Application," *Technometrics*, vol. 42, no. 2, pp. 216-217, March 2012.

[17] Resilience Engineering Association, "About Resilience Engineering," URL: *http://www.resilience-engineering-association.org/* (last accessed on February 16, 2015).