A note on the optimality of frequency analysis vs. ℓ_p-optimization

Marie-Sarah Lacharité, Kenneth G. Paterson
Information Security Group, Royal Holloway, University of London
{marie-sarah.lacharite.2015,kenny.paterson}@rhul.ac.uk

November 30, 2015

Abstract

Naveed, Kamara, and Wright's recent paper "Inference Attacks on Property-Preserving Encrypted Databases" (ACM-CCS 2015) evaluated four attacks on encrypted databases, such as those based on the design of CryptDB (Popa et al., SOSP 2011). Two of these attacks—frequency analysis and ℓ_p-optimization—apply to deterministically encrypted columns when there is a publicly-available auxiliary data set that is "well-correlated" with the ciphertext column. In their experiments, frequency analysis performed at least as well as ℓ_p-optimization for $p = 1, 2$, and 3. We use maximum likelihood estimation to confirm their intuition and show that frequency analysis is an optimal cryptanalytic technique in this scenario.

1 Overview of the attacks and attacker's capabilities

Naveed, Kamara, and Wright evaluated two attacks on deterministically-encrypted database columns when the attacker has access to an auxiliary data set [1].

Frequency analysis decrypts the column by matching the most frequent ciphertext with the most frequent plaintext from the auxiliary data (and so on for the other less frequent ciphertexts).

ℓ_p-optimization decrypts the column by matching the frequencies of the ciphertexts with the frequencies of the auxiliary plaintexts in a way that minimizes the ℓ_p-distance of their histograms.

Existing adversarial models do not describe attackers who have access to auxiliary information. The two above attacks are not ciphertext-only, since the adversary also has the auxiliary data set, nor are they known-plaintext, since the adversary does not actually know any plaintext-ciphertext pairs. Instead, they could be called *"ciphertext with frequency data"* attacks.

Decrypting a deterministically-encrypted database column using this type of auxiliary data is analogous to breaking a monoalphabetic substitution cipher given plaintext letter frequencies. However, in the context of an encrypted database column, these "letters" are not ordered, and therefore higher-order frequency statistics (of bigrams, trigrams, etc.) do not apply. Cryptanalysis of monoalphabetic substitution ciphers does not usually consider this case.

2 What is auxiliary data, exactly?

The attacker's challenge is to decrypt a deterministically encrypted column with the help of some auxiliary data. We use the language of statistics to state explicitly what *we believe* is Naveed, Kamara, and Wright's assumption: the encrypted column's underlying plaintext is a collection of independent samples of a random variable that has the distribution defined by the auxiliary data.

The auxiliary data is a multiset (i.e., which may contain repetitions) that we write as a vector $\mathbf{z} = (z_1, \ldots, z_{n_z})$, where each z_i comes from the plaintext alphabet $\mathcal{A}_M = \{m_1, \ldots, m_n\}$. Let Z be the

discrete random variable whose space is $\mathcal{A}_M$ and whose probability mass function is defined by the relative frequencies of elements in $\mathbf{z}$:

$$f_Z(m_i) = Pr(Z = m_i) = \frac{N_{m_i}}{n_z}$$

where N_{m_i} is the frequency of the symbol m_i in $\mathbf{z}$. Without loss of generality, assume that the plaintext symbols are numbered in decreasing order of frequency, i.e., $N_{m_1} \geq N_{m_2} \geq \ldots \geq N_{m_n}$.

3 Finding the most likely decryption

The encrypted database column is a multiset that we also write as a vector, $\mathbf{y} = (y_1, \ldots, y_{n_y})$, where each y_i comes from some ciphertext alphabet $\mathcal{A}_C = \{c_1, \ldots, c_n\}$. Naveed, Kamara, and Wright also assumed that the ciphertext alphabet $\mathcal{A}_C$ and the plaintext alphabet $\mathcal{A}_M$ have the same size, n. Let N_{c_i} be the frequency of the symbol c_i in $\mathbf{y}$. Frequency analysis and ℓ_p-optimization attacks rely on the assumption that the auxiliary data is "well-correlated" with the database column. We believe that Naveed, Kamara, and Wright's implicit assumption is that each y_i is a "re-labelled" sample of the random variable Z.

To decrypt $\mathbf{y}$, the adversary needs a bijective map from values in $\mathcal{A}_C$ to values in $\mathcal{A}_M$. Let π be a permutation of the integers $\{1, \ldots, n\}$. The adversary's goal is to find the permutation that maps elements of $\mathcal{A}_C$ to elements of $\mathcal{A}_M$, i.e., the π for which $c_1 = m_{\pi(1)}, \ldots, c_n = m_{\pi(n)}$.

Maximum likelihood estimation (MLE) is a technique for finding the true value of a parameter of a probability distribution function given some samples of data having that distribution.
For many classes of distributions arising naturally in applications (excepting some contrived counterexamples), MLE performs at least as well as any other statistical method for parameter estimation, for example, in the sense that it provides a minimum variance unbiased estimator for large sample sizes. Its main idea is that the parameter's true value is the one that makes the observed samples most likely. The *likelihood* of a parameter is the hypothetical probability that a particular outcome was observed given this parameter. The adversary wants to find a permutation π that maximizes the likelihood function for the observed encrypted column.

By our assumption, the samples $y_1, \ldots, y_{n_y}$ were drawn from a distribution identical to Z's, but they were "re-labelled" according to some permutation. The adversary wants to find a permutation π that was most likely to have generated $\mathbf{y}$—a permutation that maximizes $L(\pi|\mathbf{y})$, the likelihood function.

$$\begin{aligned}
&\arg\max_\pi L(\pi|\mathbf{y}) = \arg\max_\pi P(\mathbf{y}|\pi) \\
&= \arg\max_\pi \prod_{i=1}^{n_y} P(y_i | c_1 = m_{\pi(1)}, \ldots, c_n = m_{\pi(n)}) && \text{(the } n_y \text{ samples are independent)} \\
&= \arg\max_\pi \prod_{i=1}^{n} f_Z(m_{\pi(i)})^{N_{c_i}} && \text{(each } c_i \text{ appears } N_{c_i} \text{ times in } \mathbf{y} \text{ with } m_{\pi(i)}\text{'s prob.)} \\
&= \arg\max_\pi \prod_{i=1}^{n} {N_{m_{\pi(i)}}}^{N_{c_i}} && \left(f_Z(m_{\pi(i)}) = \frac{N_{m_{\pi(i)}}}{n_x}\right) \\
&= \arg\max_\pi \prod_{i=1}^{n} {N_{m_i}}^{N_{c_{\pi^{-1}(i)}}} && (\pi \text{ is a bijection})
\end{aligned}$$

Since the plaintext messages were numbered so that $N_{m_i} \geq N_{m_{i+1}}$ for each i from 1 to $n-1$, it must also be the case that $N_{c_{\pi^{-1}(i)}} \geq N_{c_{\pi^{-1}(i+1)}}$. (If $N_{c_{\pi^{-1}(i)}} < N_{c_{\pi^{-1}(i+1)}}$, then swapping their positions would yield a permutation with a strictly greater likelihood.) Therefore, the most likely permutation π is the one that assigns the most frequent plaintext in the auxiliary data to the most frequent ciphertext in the encrypted column, and so on. This permutation is simply the frequency analysis attack mentioned in the first section.

4 Conclusion

Frequency analysis naturally arises from maximum likelihood estimation of the unknown permutation. Given the power of MLE, this strongly suggests that simple frequency analysis is the "right" statistical procedure to use in our setting. Of course, in individual experiments, alternative techniques such as ℓ_p-optimization may perform better, for example, because of sampling noise. As Naveed et al. remarked, ℓ_p-optimization could also be useful when the attacker does not know which encrypted column corresponds to which auxiliary data set, because it assigns "cost information" to each pair of columns. It is interesting to note that deterministic encryption, a modern technique that allows searching on encrypted data, can be seen as a classical monoalphabetic substitution cipher with a potentially large alphabet where order does not matter.

References

[1] Naveed, M., Kamara, S., and Wright, C. V. Inference Attacks on Property-Preserving Encrypted Databases. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2015), CCS '15, ACM, pp. 644–655.