HoHo: A Hop-on Hop-off Attack Against Optimized Multiple Encoding

Ahmed Salem
ahmed.salem@cispa.de
CISPA Helmholtz Center for Information Security

Santiago Zanella-Béguelin
santiago@microsoft.com
Microsoft Research

ABSTRACT

Protocols satisfying Local Differential Privacy (LDP) allow users to protect the privacy of data they contribute for aggregation without the need to trust the aggregator. Optimal Multiple Encoding (OME) is a novel protocol presented at SIGIR 2020 to noisily encode textual data, which has been shown to enable learning from encoded text without hurting utility. OME encodes real-valued word embeddings as fixed-length binary strings and perturbs even and odd bits independently, flipping them with probabilities determined by the length of the encoding, the privacy budget ϵ, and a randomization factor λ. We show that the purported proof that OME satisfies ϵ-LDP independently of the encoding length and λ cannot hold. We present linkability and partial reconstruction attacks against OME and experimentally show that OME provides no meaningful privacy for parameter values that preserve utility. We confirm this finding using DP-Sniper, an off-the-shelf tool for discovery of differential privacy violations. For the same parameters used to experimentally show the utility of OME, DP-Sniper finds distinguishing attacks that imply that the privacy budget spent is at least 4600x higher than expected. We finally revisit the proof of LDP of OME and show that the level of privacy that it actually provides scales linearly with λ and exponentially on the encoding length, confirming our experimental findings.

KEYWORDS

machine learning, privacy, natural language processing

1 INTRODUCTION

Recent breakthroughs in natural language representation and understanding have been achieved by pre-training increasingly larger models on increasingly larger quantities of public textual data. Models pre-trained in this way still require to be fine-tuned to specific domains and tasks using high-quality sensitive data. Collecting this data poses multiple challenges, such as respecting privacy regulations and legal contracts, and reassuring data owners that no sensitive information would be leaked.

Differential privacy is the gold standard among the multiple privacy-preserving techniques that have been developed to protect users' privacy. In the distributed setting, protocols that achieve Local Differential Privacy (LDP) allow an aggegator to collate data contributed by individual users without relying on a trusted third party or expensive cryptographic techniques, such as multi-party computation and homomorphic encryption. Compared to Central Differential Privacy (CDP) where noise is added during aggregation, in LDP protocols individual users add noise before sending information to the aggregator. However, to achieve a meaningful level of privacy LDP requires adding a larger amount of noise with the subsequent decrease in utility. This often shows on CDP protocols outperforming LDP protocols in terms of the utility/privacy trade-off.

In recent work, Lyu et al. [3] address this drawback by proposing Optimized Multiple Encoding (OME), a novel LDP protocol for textual data inspired by RAPPOR [5] that achieves exceedingly good utility on classification tasks. Unlike the basic RAPPOR scheme, Symmetric Unary Encoding (SUE), and Optimized Unary Encoding (OUE) [4], but similarly to RAPPOR and other hashing-based schemes, OME uses a fixed-length binary encoding instead of a one-hot-encoding and introduces a new hyperparameter λ, called *randomization factor*. In OME, real-valued word embeddings in a sentence are encoded as bitstrings of length $\ell = m + n + 1$, using m bits for the integral part, n bits for the fractional part and 1 bit for the sign, and perturbed independently. A given text of maximum length r is thus padded and encoded as a bitstring of length $r\ell$. The randomization factor λ determines the probability with which individual bits are flipped. This probability depends on whether a bit is at an even or and odd position in an encoded embedding, and similar to OUE, on both whether the bit is set or not.

Surprisingly, Lyu et al. [3] show that the privacy guarantee of OME is independent of λ and the representation length $r\ell$. In this paper, we scrutinize this purported privacy guarantee. We start by developing a Hop-on Hop-off (HoHo) attack, which exploits the difference with which each bit is flipped introduced by λ to break privacy of text encoding with OME. We then revisit the proof of LDP privacy of OME, identify a flaw and fix it by revising the privacy bound. Our revised proof shows that indeed, the privacy of OME depends linearly on λ and exponentially on the representation length, so that the influence of λ outweighs other factors.

Finally, we confirm our findings using DP-Sniper [1], a state-of-the-art black-box detector for differetial privacy violations. The results of DP-Sniper confirm our findings and show the high dependency on λ of the OME privacy guarantee.

2 OPTIMIZED MULTIPLE ENCODING (OME)

In this section, we start by presenting the local differential private (LDP) protocol proposed by Lyu et al. [3], namely the Optimized Multiple Encoding (OME), then present its proof.

2.1 OME Protocol

OME aims at generating differentially private text representation while preserving enough utility to train state-of-the-art Natural language Processing (NLP) based models.

Privacy preserving machine learning (PPML) workshop at ACM CCS 2021, Seoul, South Korea

The OME mechanism introduces two main changes to the current state-of-the-art ϵ-LDP protocols, i.e., Symmetric Unary Encoding (SUE) [5] and Optimized Unary Encoding (OUE) [4]. Firstly, instead of using one-hot encoding to encode the inputs (in this case, the text embeddings), they encode the values into binary vectors with length l. Thus, for an embedding with r elements, the sensitivity of OME is rl. Secondly, OME follows the OUE mechanism of perturbing bits with the values 0 and 1 with different probabilities. However, it further improves the encoded vectors utility by introducing a hyperparameter that further controls the probability of perturbing the bits depending on their locations, namely λ. Intuitively, increasing λ increases the probability of preserving the even bits of the vector and decreases the one for the odd bits. Hence, the OME mechanism should preserve both utility and privacy. More formally, in [3] the authors present the following theorem that formalizes the guarantees of the OME:

Theorem 2.1. *For any inputs v, v' and any encoded bit vector B with sensitivity rl, OME provides ϵ-LDP given*

$$p = \Pr\{1 \rightarrow 1\} = \begin{cases} \frac{\lambda}{1+\lambda}, \textit{for } i \in 2n \\ \frac{1}{1+\lambda^3}, \textit{for } i \in 2n+1 \end{cases} \quad (1)$$

$$q = \Pr\{0 \rightarrow 1\} = \frac{1}{1+\lambda e^{\frac{\epsilon}{rl}}} \quad (2)$$

2.2 OME Proof

In [3] the authors present the following proof to prove the previously presented theorem (Theorem 2.1).

Proof. Let v and $\vec{B}$ represent an input and its encoded bit representation. Given that $\vec{B}$ has a sensitivity of rl, the privacy budget ϵ needs to be divided by the sensitivity for each bit. By setting

$$p = \Pr\{1 \rightarrow 1\} = \begin{cases} \frac{\lambda}{1+\lambda}, \text{for } i \in 2n \\ \frac{1}{1+\lambda^3}, \text{for } i \in 2n+1 \end{cases} \qquad q = \Pr\{0 \rightarrow 1\} = \frac{1}{1+\lambda e^{\frac{\epsilon}{rl}}}$$

$$1-p = \Pr\{1 \rightarrow 0\} = \begin{cases} \frac{1}{1+\lambda}, \text{for } i \in 2n \\ \frac{\lambda^3}{1+\lambda^3}, \text{for } i \in 2n+1 \end{cases} \qquad 1-q = \Pr\{0 \rightarrow 0\} = \frac{\lambda e^{\frac{\epsilon}{rl}}}{1+\lambda e^{\frac{\epsilon}{rl}}}$$

Then for any inputs v, v', we have

$$\frac{\Pr\{\vec{B}|v\}}{\Pr\{\vec{B}|v'\}} = \frac{\prod_{i=1}^{rl}\Pr\{B[i]|v\}}{\prod_{i=1}^{rl}\Pr\{B[i]|v'\}} = \frac{\prod_{i\in 2n}\Pr\{B[i]|v\}}{\prod_{i\in 2n}\Pr\{B[i]|v'\}} \times \frac{\prod_{i\in 2n+1}\Pr\{B[i]|v\}}{\prod_{i\in 2n+1}\Pr\{B[i]|v'\}}$$

$$\leq \left(\frac{\Pr\{1\rightarrow 1\}}{\Pr\{1\rightarrow 0\}} \times \frac{\Pr\{0\rightarrow 0\}}{\Pr\{0\rightarrow 1\}}\right)^{\frac{rl}{2}}_{i\in 2n} \times \left(\frac{\Pr\{1\rightarrow 1\}}{\Pr\{1\rightarrow 0\}} \times \frac{\Pr\{0\rightarrow 0\}}{\Pr\{0\rightarrow 1\}}\right)^{\frac{rl}{2}}_{i\in 2n+1}$$

$$= \left(\frac{\frac{\lambda}{1+\lambda}}{\frac{1}{1+\lambda}} \times \frac{\frac{\lambda e^{\frac{\epsilon}{rl}}}{1+\lambda e^{\frac{\epsilon}{rl}}}}{\frac{1}{1+\lambda e^{\frac{\epsilon}{rl}}}}\right)^{\frac{rl}{2}} \times \left(\frac{\frac{1}{1+\lambda^3}}{\frac{\lambda^3}{1+\lambda^3}} \times \frac{\frac{\lambda e^{\frac{\epsilon}{rl}}}{1+\lambda e^{\frac{\epsilon}{rl}}}}{\frac{1}{1+\lambda e^{\frac{\epsilon}{rl}}}}\right)^{\frac{rl}{2}} = e^{\epsilon}$$

□

3 REVISITING THE OME

In this section, we first present our intuition and technique of our HoHo attack against the OME mechanism. Next, we revisit the OME's original proof to compare it against our results.

3.1 The Hop-on Hop-off Attack

Since the OME mechanism provides a better utility by mainly maintaining the even bits with a high probability, we believe that an adversary can utilize this information presented in the even bits to break the OME. To this end, we present our Hop-on Hop-off (HoHo) attack, where the adversary selectively considers a subset of the bits, i.e., the adversary focus (hops on) the even bits and ignores (hops off) the odd ones. More concretely, for an encoded sentence s and a plaintext sentence t, we first calculate the embedding of t without any perturbation. Next, we compute a matching score by comparing the values of even bits of s and the embedded version of t. The larger the matching score, the higher the probability that this encoded sentence s is the output of the OME when encoding the plaintext sentence t. We present our HoHo attack in Algorithm 1.

We follow [2] and explore different settings where the adversary can use the HoHo attack to infer different information from the OME's encodings. We briefly discuss these settings below:

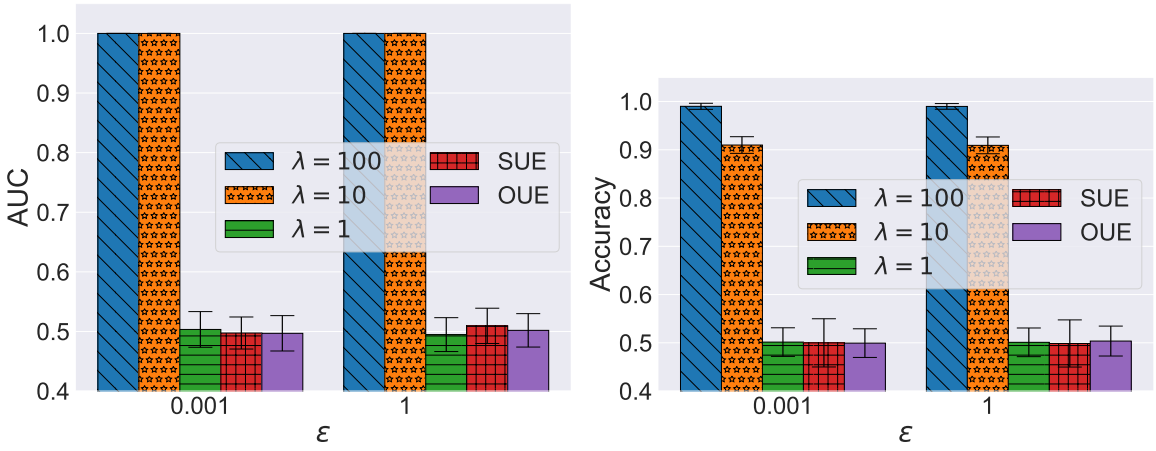
(1) *Distinguishability:* The first setting we consider is distinguishability. In this setting, an adversary aims at distinguishing if a given encoding corresponds to a specific sentence or not. For instance, the adversary can give a challenger two sentences with the same concept function [2]. The challenger then randomly picks one of these sentences and encodes it using OME. Next, the challenger sends the encoding to the adversary. Given the encoding, the adversary tries to determine which sentence was encoded. The adversary wins this game if they predict the correct encoded sentence.
(2) *Linking:* In the second setting, namely Linking, we try to generalize the first one. More concretely, instead of picking two sentences with the same concept. The adversary is given a list of different encodings and a target sentence. The adversary wins if they can link the target sentence to its corresponding encodings inside the given list. Constructing this list of encodings can relax the assumptions needed for the concept function used in the distinguishability setting. For instance, using the encoding of the complete dataset in addition to multiple occurrences of the target sentence in the encoding list ensures that there exist multiple sentences with the same concept as the target sentence.
(3) *Reconstruction:* Finally, the most complex setting is the reconstruction one. In this setting, the adversary tries to reconstruct the original sentence/embedding given the encoding.

In this paper, we consider the two most complex settings, namely Linking and Reconstruction.

Linking: We first evaluate our HoHo attack in the linking setup. To this end, we construct the list of embeddings by encoding the target sentence 100 times and mixing these encodings with the encodings of the complete dataset. Next, we use our HoHo attack to calculate the score between each encoding and the target sentence embedding. We set the labels to be 1 for the target sentences and 0 otherwise. Finally, we calculate the AUC score for the different scores and their corresponding labels.

To implement this attack, we use the code published by the authors for the OME algorithm [1] [3], and implement our attack using

[1] https://github.com/lingjuanlv/Differentially-Private-Text-Representations



(a) Linking Attack **(b) Reconstruction Attack**

Figure 1: The results of the linking (Figure 1a) and reconstruction (Figure 1b) settings using our HoHo attack for different values of ϵ when using OME (with multiple values of λ), SUE and OUE. As the figure shows, independent of the value of ϵ, our attack can achieve a perfect AUC score for the large values of λ.

python. We will publish the code of our attack for reproduction. We follow the authors and use the yelp dataset for evaluating our HoHo attack. We sample 800 random reviews to create our dataset and randomly sample 10% for the target sentences. Finally, we perform the HoHo attack in the linking setting independently on each target sentence and take the average – and standard deviation – for the different runs.

We evaluate our attack using two ϵ values, i.e., 0.001 and 1, to show its effectiveness. For each of the ϵ values, we evaluate the OME, SUE, and OUE mechanisms. Moreover, for the OME mechanism, we try different values for λ, namely 1, 10, and 100. Finally, we plot our attack results in Figure 1a.

As the figure shows, independently of the ϵ used, when the value of λ is high, i.e., 10 or more, our HoHo attack can achieve a perfect AUC of 1.0. Dropping the λ to 1, makes the performance of the OME – in terms of privacy – similar to the one of SUE and OUE. However, it is important to mention that with low values of λ, the OME loses its advantage of better utility compared to the other two mechanisms (OUE and SUE) [3].

Reconstruction: Second, we evaluate the reconstruction setting. In this setting, we aim at reconstructing the even bits of the encoding. To this end, we reconstruct the even bits by taking the ones of the encoding as is. We use the same evaluation setting as the one previously introduced in the Linking setting, except for using the accuracy to measure the performance of our attack, i.e., we calculate accuracy between the reconstructed bits and the original ones.

Figure 1b plots the results of our reconstruction attack. As expected, the higher the value of λ, the more successful our attack is. For instance, our attack is able to achieve 99% accuracy for $\lambda = 100$, for both values of ϵ. The performance drops to 91% when λ drops to 10. Similar to the linking settings, when λ is equal to 1, the performance of the OME achieves approximately random guessing accuracy (50%) as the SUE and OUE mechanisms.

Input: encoded sentence s, plaintext sentence t
Output: the matching score acc
$e_t = embed(t)$ *//Calculating the embedding of t*
$acc = 0$ *//Initializing the matching score*
for $i = 0$ **to** $len(e_t)$ **do**
  //both vectors e_t and s have the same length
  if $i\%2 == 0$ **then**
    //This is an even bit
    $acc = acc + (e_t[i] == s[i])$
  end if
end for
//Normalizing the matching score
$acc = acc/floor(len(e_t)/2)$

Algorithm 1: The Hop-on Hop-off Attack

The results of this and the linking settings clearly demonstrate the privacy leakage of the OME with high values of λ. Moreover, they show that the privacy leakage of OME is highly dependant on the value of λ.

3.2 Revisiting The Proof

As previously demonstrated, our results show that the OME's privacy guarantees depend on λ in addition to ϵ, i.e., the higher the λ, the better our attacks are. Hence, we revisit the proof previously presented in Section 2.2, as it shows that the privacy guarantees are independent of λ.

We believe the proof (Section 2.2) does not assume the worst case when performing the analysis (the second line of the proof in Section 2.2). Thus, we now present the modified proof with what we believe is the worst case:

Proof. For any inputs inputs v, v', and output $\vec{B}$ we have:

$$\frac{\Pr\{\vec{B}|v\}}{\Pr\{\vec{B}|v'\}} = \frac{\prod_{i=1}^{rl}\Pr\{B[i]|v\}}{\prod_{i=1}^{rl}\Pr\{B[i]|v'\}} = \frac{\prod_{i\in 2n}\Pr\{B[i]|v\}}{\prod_{i\in 2n}\Pr\{B[i]|v'\}} \times \frac{\prod_{i\in 2n+1}\Pr\{B[i]|v\}}{\prod_{i\in 2n+1}\Pr\{B[i]|v'\}} \tag{3}$$

$$\frac{\prod_{i\in 2n}\Pr\{B[i]=1|v\}}{\prod_{i\in 2n}\Pr\{B[i]=1|v'\}} \times \frac{\prod_{i\in 2n+1}\Pr\{B[i]=0|v\}}{\prod_{i\in 2n+1}\Pr\{B[i]=0|v'\}} = \frac{\prod_{i\in 2n} p}{\prod_{i\in 2n} 1-p} \times \frac{\prod_{i\in 2n+1} 1-q}{\prod_{i\in 2n+1} q} =$$

$$\frac{\prod_{i\in 2n}\frac{\lambda}{1+\lambda}}{\prod_{i\in 2n} 1-\frac{\lambda}{1+\lambda}} \times \frac{\prod_{i\in 2n+1} 1-\frac{1}{1+\lambda e^{\frac{\epsilon}{rl}}}}{\prod_{i\in 2n+1}\frac{1}{1+\lambda e^{\frac{\epsilon}{rl}}}} = \frac{\prod_{i\in 2n}\frac{\lambda}{1+\lambda}}{\prod_{i\in 2n}\frac{1}{1+\lambda}} \times \frac{\prod_{i\in 2n+1}\frac{\lambda e^{\frac{\epsilon}{rl}}}{1+\lambda e^{\frac{\epsilon}{rl}}}}{\prod_{i\in 2n+1}\frac{1}{1+\lambda e^{\frac{\epsilon}{rl}}}} =$$

$$\prod_{i\in 2n}\lambda \times \prod_{i\in 2n+1}\lambda e^{\frac{\epsilon}{rl}} = \lambda^{\frac{rl}{2}} \times (\lambda e^{\frac{\epsilon}{rl}})^{\frac{rl}{2}} = \lambda^{\frac{rl}{2}} \times \lambda^{\frac{rl}{2}} e^{\frac{\epsilon}{2}} = \lambda^{rl} e^{\frac{\epsilon}{2}} \tag{4}$$

□

Equation 3 is due to bits being encoded/flipped independently. Equation 4 is because this probability is maximized when the even bits of v and $\vec{B}$ are set to 1 and the odd bits to 0, and v' to the inverse of them.

This confirms our intuition that the OME's privacy guarantees indeed depend on the value of λ.

4 DP SNIPER

To further confirm our findings, we use a state-of-the-art technique for automatically finding the differential privacy violations. Namely, we the DP-Sniper [1], which provides a black-box tool for the detection of such violations.

We again use the code published by the authors, however, we simplify the following for DP-Sniper:

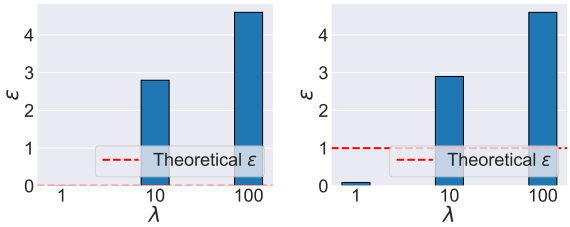
(1) Instead of encoding a vector of floating numbers, we simplify it to a single floating number.
(2) We limit the input to the OME mechanism to be any floating number from -10 to 10.

With those simplifications, we run the DP-Sniper for multiple values of ϵ (0.001 and 1) and λ (1, 10, and 100), and plot the results in Figure 2.

Figure 2a and Figure 2b shows the DP-Sniper results for the $\epsilon = 0.001$ and $\epsilon = 1$ cases, respectively. As expected, the practical ϵ clearly exceeds the theoretical one for large values of λ. For instance, the estimated ϵ for the OME mechanism with $\lambda = 100$ exceeds the theoretical one with a factor of 4,600 and 4.6 when setting the theoretical ϵ to 0.001 and 1, respectively. Decreasing λ to 10 reduces the estimated ϵ, however, it still exceeds the theoretical one with a factor of 2,900 and 2.8 for both values of theoretical ϵ (0.001 and 1). Finally, similar to our previous findings, settings $\lambda = 1$ satisfies the ϵ-LDP guarantees.

The results of the DP-Sniper confirms our findings:

(1) The OME privacy guarantees depend on the value of λ.
(2) For high values of λ, the OME mechanism leaks information above the expected guarantees.
(3) The influence of λ on the privacy of the OME mechanism tremendously outweighs the one of ϵ.



(a) $\epsilon = 0.001$ **(b)** $\epsilon = 1$

Figure 2: The results of running DP-Sniper against the OME mechanism with different values of ϵ. Figure 2a and Figure 2b shows the results for $\epsilon = 0.001$ and $\epsilon = 1$, respectively. We plot the theoretical ϵ (the red dotted line) to compare the privacy leakage of the different settings. An ϵ-LDP mechanism should have practical ϵ lower than the theoretical one, i.e., it should be below the red line. As both figures show, using high values of λ can lead to significant privacy leakage.

5 CONCLUSION

In this paper, we present the Hop-on Hop-off (HoHo) attack against the state-of-the-art differentially private text representations, namely the Optimized Multiple Encoding (OME). Our attack achieves a strong performance for the different values of ϵ. We revisit the OME proof and propose a new one that proves the dependency of the OME's privacy guarantees on λ. Finally, we use an off-the-shelf state-of-the-art differential privacy violation detector, namely DP-Sniper, to validate our findings and show the privacy leakage of the OME mechanism.

REFERENCES

[1] Benjamin Bichsel, Samuel Steffen, Ilija Bogunovic, and Martin Vechev. 2021. DP-Sniper: Black-Box Discovery of Differential Privacy Violations using Classifiers. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE.
[2] Nicholas Carlini, Samuel Deng, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, Shuang Song, Abhradeep Thakurta, and Florian Tramèr. 2021. Is Private Learning Possible with Instance Encoding?. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE.
[3] Lingjuan Lyu, Yitong Li, Xuanli He, and Tong Xiao. 2020. Towards Differentially Private Text Representations. In *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1813–1816.
[4] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. Locally Differentially Private Protocols for Frequency Estimation. In *26th USENIX Security Symposium, USENIX Security 2017*. USENIX Association, 729–745.
[5] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 21st ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1054–1067.