# i-SEAL$^2$: Identifying Spam EmAiL with SEAL

I. Demertzis, D. Froelicher*, N. Luo, M. Norberg Hovd

## Introduction

End-to-end encrypted emails are desirable with regards to privacy, as it prevents your email provider from storing and reading your emails in plaintext. However, with the perk of privacy from the end-to-end encryption, you lose the spam filter, as the filtering process requires an analysis on the email's content, or its metadata. The classification of whether an email is spam typically relies on machine learning algorithms that have been trained on large amounts of emails.

A naive approach to combine end-to-end encryption of emails and a spam filter would be for every user to simply build their own model using only their own emails to train the machine learning model. However, one user typically only has a limited number of emails and this local approach is going to result in a model which is less accurate than the one provided by an email provider, simply due to the size of the dataset used to train the machine learning model. In order to obtain an accurate model, large amounts of diverse data are required.

Another approach would consist in sharing only spam emails, which usually do not contain sensitive or private information, with the email provider. This approach solves the previous problem of having a too small training set. However, it would result in a one-class classification scenario, as the machine learning model would be trained on distinguishing spam and ham emails by being trained on a set containing mostly spam emails. This will result in a less accurate model than one trained on a dataset with an even (or less biased) distribution of both types of emails.

In this short paper, we suggest to rely on homomorphic encryption to circumvent this problem, and propose a solution that enables privacy-preserving classification of emails as spam or ham. We also describe a solution for an oblivious training of spam detection machine learning model that enables the training of accurate detection model without hindering the privacy of the users.

## Private Classification

In Figure 1.A, email users want to obtain a classification of their emails as spam or ham. Email providers, e.g., Google or Microsoft, already have (cleartext) classification models, which are trained on millions of users' emails and can be used to perform this classification. However, in today's solution, this requires them to access the private content of the emails.

In order to avoid this, we propose a solution in which the user sends encrypted information, generated from the received emails, i.e., a vector of encrypted features, to the service provider and receives in return a (encrypted) classification of the email. This is executed without revealing the content of the emails to the service provider.

When receiving an encrypted email, the user decrypts it and (automatically) generates a vector of features capturing its content. This features' vector is then encrypted under the client's public key and sent to the email provider, who classifies the email as spam or ham by using its cleartext model on the received encrypted feature vector. The provider then sends the encrypted classification back to the client, who decrypts it using its secret key and obtains an automatic and privacy-preserving classification of his emails.

*corresponding author, david.froelicher@epfl.ch

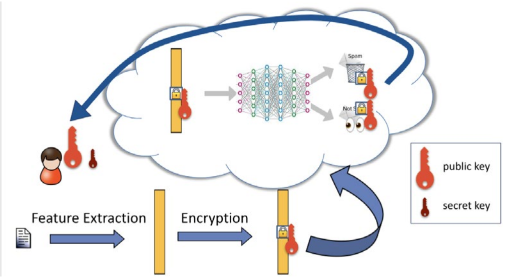

Figure 1.A: Private Classification

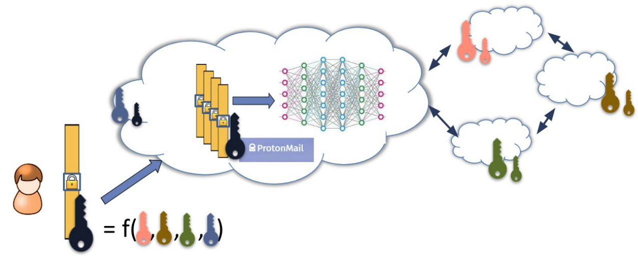

Figure 1.B: Private Training

## Private Training

While private classification enables users to benefit from the automatic spam detection without hindering their privacy, it also results in service providers not obtaining more data in order to train and update their detection model. To counter this issue, we propose a solution that enables the service providers to train their model on encrypted data.

As presented in Figure 1.B, email users send a feature vector of their email encrypted under the collective public key of the service providers. This key is a combination of the providers' public keys and ensures that the decryption of a ciphertext under the collective public key is only possible when all the service providers collaborate in the decryption, by using their own secret key. The encrypted material will only be decrypted once all the service providers have "partially" decrypted by using their respective secret keys.

The service provider trains and periodically updates an encrypted model for spam detection by using the encrypted vectors collected from the email users.

The private classification can then be performed as presented before, using the updated model. We observe here that this model can be periodically decrypted such that the service provider can use the cleartext model to perform the classification on encrypted data. Alternatively, at the end of every period, the updated (and decrypted) model is sent to the users, which can perform the email classification locally.

We also note that users may choose to disclose emails in plaintext to the provider for training purposes. This simplifies the continuous model training, as it is partially executed on cleartext data, thus reducing the computation complexity. With this alternative, one may train the model on plaintext spam and non-sensitive ham emails, and use homomorphic encryption for training the model only on sensitive emails, where the users themselves choose which emails are too sensitive to share in plaintext.

## Conclusion

We propose a system that enables both the privacy-preserving classification of emails as spam or ham and the secure training of the classification model, thus providing a solution that responds to the growing usage of end-to-end encrypted emails and the increasing demand for privacy-preserving solutions.