Automatic detection of weak cipher usage in aircraft communications

Francesco Intoci, Mattia Mariantoni, Theresa Stadler
Department of Computer Science, EPFL, Switzerland
-
Supervised by Kasra Edalatnejadkhamene
SPRING Lab, EPFL, Switzerland

***Abstract*—The Aircraft Communications Addressing and Reporting System (ACARS) allows aircraft to communicate with entities on the ground via short messages. To provide confidentiality for sensitive information communicated via the ACARS network, some operators started to deploy proprietary cryptography to encrypt message contents. This is highly problematic, however, as all of the observed approaches in practice offer next to no communication security but give a false sense of it. Authorities hence would like to filter out weak ciphers at the network level to alert operators of their risks.**
This project explored the use of deep convolutional neural networks (CNN) for automatic detection of weakly encrypted message contents on the ACARS data link network. We constructed a labelled dataset of plaintext and ciphertext messages and experimentally evaluated the performance of a deep CNN for message classification. We find that a model trained on a sufficiently diverse set of ciphertexts is able to detect weakly encrypted messages with high accuracy.

I. Introduction

The Aircraft Communications Addressing and Reporting System (ACARS) [1] builds the foundation of modern data communication between aircraft and entities on the ground. Through ACARS, aircraft transmit and receive short messages with vital information about flight plans, weather information, equipment health, etc. The content of ACARS messages ranges from sensor data automatically generated by an aircraft's internal systems to user-generated communications between aircraft crew members and ground stations. Messages are encoded as a sequence of basic ASCII characters and use a variety of message formats and protocols.

A growing concern about the widespread use of the ACARS data link system is that the vast majority of communications is still *submitted in the clear and offers no confidentiality*. This problem has come to the attention of privacy and security researchers due to two relatively recent developments [2]. First, while originally used primarily by commercial airlines, the ACARS network nowadays serves as a multi-purpose data communication link system for many entities including state actors and the military. This development significantly changes the sensitivity of the transmitted content. Second, until recently, specialist hardware and software was required to receive ACARS messages. The introduction of software-defined radios and the development of custom software to decode aircraft communications, however, today enables any sufficiently tech-savvy user to intercept ACARS-transmitted messages. This development increases the risk of eavesdropping attacks as potential adversaries need less resources and specialist knowledge to intercept communications [2].

A recent measurement study on the privacy of sensitive content submitted on the ACARS channel found a significant information leakage [2]. The study demonstrates that ACARS messages submitted in the clear undermine efforts to hide sensitive location information and cause major privacy issues to business, military, and government aircraft.
Many of the described issues are a result of cleartext message contents being freely accessible to passive eavesdroppers. The study thus *recommends the use of standardised cryptographic approaches* to protect the confidentiality of message contents. While official systems, such as the Secure ACARS message standard [3], are available, a survey finds that, in practice, deployment is close to non-existent [2]. Instead, some operators *use non-standardised, proprietary cryptography* to improve the confidentiality of message contents. However, all the proprietary approaches identified so far rely on *insecure (substitution) ciphers* and provide no meaningful level of security [2]. This is a concerning observation. Weak encryption provides an *illusion of security* and detracts attention from developing standardised, secure solutions that provide real confidentiality for privacy sensitive message contents.

Task description. To combat the use of weak encryption in the ACARS data link network, a first step would be to *identify and flag messages encrypted under weak ciphers*. This would enable authorities to filter out such messages at the network level and to alert operators of the usage of weak ciphers by their aircraft.

The goal of this project was to explore the use of deep convolutional neural networks (CNNs) for automatic detection of weakly encrypted message contents on the ACARS data link network. We cast this problem as a supervised learning task in which a classifier was trained to label the content of a message transmitted via the ACARS channel as either plaintext or a message encrypted under an insecure cipher. Similar to previous text classification tasks, in which a free text document is assigned to one out of a fixed set of predetermined categories [4], we treated the content of each message as

```
plain_manual
QUTLDAA011171FTM\nMARK HAD U PLANNED FL300 FOR\nSOME FCST CHOP
AFTER 5050N\IF HIGHER BETTER ID STAY WITH IT\nFD53 JON DOE\nPLZ
ACK MSG NO. (7717)\n END

plain_sensor
- #MD/AA VLKKPYA.AK1.N844MH60A8A42101L9CB

cipher_columnar
NA 57H3TKA2B53AI-C/PCA49YA08AM. 85 BFE.1DAM0-E#2
```

Fig. 1. Examples of the three classes of message contents observed in the ACARS corpus. Data slightly modified to prevent sensitive information leakage.

a sequence of characters with a label (`plain` or `cipher`) attached to it.

Challenges. Distinguishing plaintext messages from those encrypted under insecure ciphers, however, is a challenging task due to several reasons.

First, in comparison to the text documents written in a unified natural language found in traditional text classification tasks, the content of ACARS messages is highly variable and ranges from machine-generated sensor data to manually typed messages written in multiple languages. A manual inspection of example messages from the ACARS corpus shows that the structure of the text differs significantly from ordinary texts (Examples shown in Fig. 1). Messages are not composed of words but are rather a sequence of characters drawn from the full set of basic ASCII characters. User-generated texts contain words from a mix of languages.

Second, ACARS messages that contain aircraft-generated sensor data seem to be indistinguishible from the random stream of characters produced by some ciphers. This makes it challenging to build a robust classifier that differentiates between these two classes.

Last, the data available through data collection of real-world communications on the ACARS network is not naturally labelled and might contain ciphertexts. This requires careful consideration when pre-processing and cleaning the raw data for classification.

Contributions. In short, we implemented and evaluated the performance of a character-level CNN for automatically detecting the use of weak ciphers to encrypt message contents in the ACARS data link network. We constructed a large corpus of messages encrypted under weak substitution ciphers and used the generated data to train a deep network model to distinguish these messages from their plaintext counterparts. We find that the model provides a highly accurate classifier and is able to distinguish plaintext from encrypted messages with up to 99.96% accuracy.

II. Data Description and Processing

A dataset of $\sim$1M real-world messages collected from the ACARS data link network was provided. Each record contained the original message content, a variable length sequence of ASCII characters, and associated metadata. The metadata included information about the channel on which the message was submitted, for instance, via satellite or vhf, and a message tag that indicates the interface from which the message was sent, for instance, by internal aircraft systems or through a manual user interface by aircraft crew members. This information was used to filter and preprocess the data.

As the main goal of this project was to distinguish messages encrypted under weak ciphers from cleartexts, the message tags were used to select all records that, according to the official ACARS specification [1], would likely contain either plaintext messages written by aircraft crew or sensor data. The tags `RA` and `C1` were found to most likely contain manually sent messages authored by crew members. The tag `H1` indicates with high likelihood a message transmitting sensitive sensor data from an aircraft's internal systems. Furthermore, messages submitted via satellite were most likely to contain message contents sent in the clear.

For our initial evaluation, we hence only considered messages sent via satellite and broadly categorised these messages as either a manually sent plaintext message (label: `plain_manual`) or a plaintext message containing sensor data (label: `plain_sensor`) depending on their tag. The metadata was only used for the initial data preprocessing and was not taken into account as a feature for classification.

Ciphertext generation. The dataset obtained through this initial pre-processing step contained $93,132$ message texts labelled as `plain_manual` and $178,914$ texts labelled as `plain_sensor`. Both of these types, however, represent examples of the negative class (`plain`) in the final classification task. To obtain a comprehensive train and test set, we generated encrypted versions of these messages with the following set of simple ciphers:

Ceasar cipher. The Caesar cipher, also referred to as shift cipher, is a simple substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet, i.e., each letter is "shifted" by a fixed number of positions [5]. †The size of the shift is determined by the key of the cipher.

Substitution cipher. The simple substitution cipher differs from the Caesar cipher in that the cipher alphabet is not simply the alphabet shifted. Each plaintext character is substituted with a random character from the plaintext alphabet [5]. The key of the simple substitution cipher hence consists of all characters in the alphabet presented in a random order.

Vigenere cipher. The vigenere cipher is a polyalphabetic substitution cipher that uses the Caesar cipher as one of its elements. A keyword in combination with a "tabula recta" is used to encode each character in the plaintext [5].

Columnar cipher. The columnar cipher is a transposition cipher in which plaintexts are written out in rows of a fixed length and ciphertexts are obtained by reading out the texts column by column in a scrambled order [5]. The width of rows and the reading order of columns is usually determined by a keyword.

All of these ciphers offer essentially no communication security and can be easily broken with little computational resources (some even by hand).

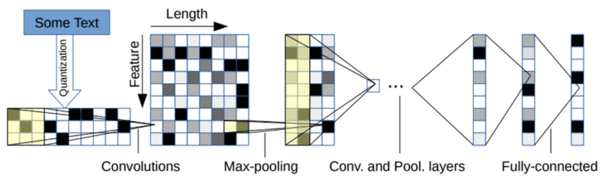

Fig. 2. Character-level convolutional network by Zhang et al. [7]

III. Character-level Convolutional Neural Networks for Text Classification

As outlined in Section I, the project's goal was to explore whether it is possible to train a classifier h that given the content of a message from the ACARS data link network $\mathbf{m}$ labels this message as either plaintext (`plain`) or encrypted under a weak cipher (`cipher`). A message $\mathbf{m} = (m_1, m_2, \cdots, m_l)$ is a sequence of characters from the basic ASCII character set and the task of assigning a sequence of text characters to predefined categories is commonly referred to as text classification [4].

Message encoding. While almost all text classification models use the natural construct of words to quantise raw text data into numerical features [4], [6], in this work, we follow the approach presented by Zhang et al. [7] and treat each message as a sequence of characters. This approach has multiple advantages and addresses many of the challenges described in Section I. Modelling a message as a 1-dimensional signal at character-level makes the classification agnostic to the language in which a message was written, works on abnormal character sequences, such as those produced by internal aircraft systems, and enables the use of CNNs which have been found useful in extracting information from sequential data.

Following the approach presented by Zhang et al. [7], we encode each message $\mathbf{m}$ as a fixed length sequence of one-hot vectors. Given an alphabet of size k, each character m_i is translated into a vector of size $k \times 1$ which is all zeros except at position k_i which indicates the position of character m_i in the alphabet. Starting from the last position, a message $\mathbf{m}$ is thus transformed into a binary array of size $k \times l$ where l is the number of characters in the message. We fix the maximum sequence of length of any message to l_{max} and ignore any characters that exceed this limit. Messages with $l < l_{max}$ are zero-padded to the maximum sequence length.

Model architecture. The model described by Zhang et al. [7] is a 9 layers deep CNN with 6 convolutional and 3 fully-connected layers. We implement the small version of the model with 256 1-dimensional kernels in each convolutional layer and varying kernel size (see original paper and implementation for details [7], [8]). The first two and the last convolutional layers are followed by temporal max-pooling which enables training of deeper models [7]. The first two fully-connected layers present a dropout layer with dropout probability $p = 0.5$. The number of input features to the first convolutional layer and the number of output features of the last fully-connected layer are determined by the problem. In our case, the alphabet size is equal to the full set of ASCII characters 128 and number of classes is either equal to 2 or 3 depending on the classification problem (see Section V).
We use the same non-linearity (ReLU) and training procedure (mini-batch SGD) as in the original paper.

Implementation. The model was implemented under the `PyTorch` framework [9]. All code and documentation can be found online [8]. The implementation provides functions for training and evaluating a trained model and can be adapted to run either the small or large version of the CNN presented by Zhang et al..

IV. Experimental Setup

In this section, we described the experimental setup used to evaluate model performance on the dataset described in Section II.

Datasets. The set of messages labelled as `plain_manual` was split into a train $\mathtt{M}^{\mathtt{train}}$ and test set $\mathtt{M}^{\mathtt{test}}$ with a ratio of 75% to 25%. These two sets were used to generate various combinations of training and evaluation sets.

We first apply the ciphers described in Section II to all plaintext messages in the training set $\mathtt{M}^{\mathtt{train}}$ keeping the key of each cipher fixed to obtain the train set named $\mathtt{M}^{\mathtt{train}}_{\mathtt{fixed}}$. The same set of ciphers *using the same set of keys as for the training set* is then used to encrypt all messages in the test set. We label this test set $\mathtt{M}^{\mathtt{test}}_{\mathtt{fixed}}$. We then rotate the key of each cipher and generate a second test set $\mathtt{M}^{\mathtt{test}}_{\mathtt{rot}}$ that contains both the plaintext messages contained in $\mathtt{M}^{\mathtt{test}}$ and the encrypted versions. Last, we generate a second training set $\mathtt{M}^{\mathtt{train}}_{\mathtt{rot}}$ by encrypting all plaintext messages in $\mathtt{M}^{\mathtt{train}}$ this time rotating the key for each cipher frequently (every 1000 messages).

The same procedure was repeated on all plaintext messages labelled as `plain_sensor`. We refer to these sets as $\mathtt{S}^{\mathtt{train}}$ and $\mathtt{S}^{\mathtt{test}}$, respectively.

Model training. On each of the resulting training sets, we trained multiple versions of the model described in Section III. We varied the optimisation procedure, batch size, and maximum sequence length used for training and obtained the best results with the same hyperparameter settings as described by Zhang et al. [7].

Performance evaluation. Ideally, a trained model should be able to distinguish plaintexts from ciphertexts with high *accuracy*. In addition, we report the observed *false negative rate* of each model on different test sets, i.e., the percentage of ciphertext messages wrongly labelled as plaintext. A high false negative rate indicates that the trained model would be likely to miss messages encrypted under insecure ciphers. This would weaken the usefulness of the approach in practice.

V. Results

In this section, we present the results of our experimental evaluation. Across our experiments, we varied different aspects

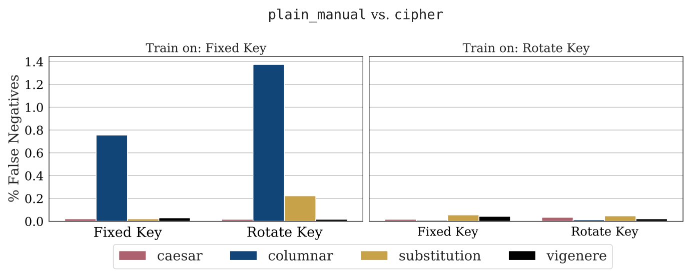

Fig. 3. *Left*: False negative rate of a model trained on $\mathtt{M}^{\mathtt{train}}_{\mathtt{fixed}}$ evaluated over $\mathtt{M}^{\mathtt{test}}_{\mathtt{fixed}}$ and $\mathtt{M}^{\mathtt{test}}_{\mathtt{rot}}$ *Right*: False negative rate on the same test sets for a model trained on $\mathtt{M}^{\mathtt{test}}_{\mathtt{rot}}$.

of the model training set to explore the following set of research question.

Can the model distinguish manually sent plaintext messages from weak substitution ciphers? We first evaluated the performance of a model trained on $\mathtt{M}^{\mathtt{train}}_{\mathtt{fixed}}$ on a test set $\mathtt{M}^{\mathtt{test}}_{\mathtt{fixed}}$ that contained a set of hold-out plaintext messages and ciphertexts from the same ciphers (fixed key) as those that were used to generate the training examples. We find that in this setup the model is able to distinguish between the two classes `plain_manual` and `cipher` with 99.85% accuracy. In comparison, a simple baseline model that models each message as a sequence of characters drawn i.i.d. from the marginal character distribution found in the training data performs only slightly better than random guessing (53.94% accuracy on the test set)[1].

Does key rotation affect classification accuracy? In Fig. 3 *Left*, we compare the performance of a model trained on ciphers with a fixed key on two different test sets: The test set $\mathtt{M}^{\mathtt{test}}_{\mathtt{fixed}}$ in which ciphers share a key with the model's training set and the test set $\mathtt{M}^{\mathtt{test}}_{\mathtt{rot}}$ in which a cipher's key differs from the training set. We find that while overall classification accuracy remains high (99.72%) there is a slight increase in the false negative rate for messages encrypted under the columnar cipher if the cipher key is rotated between train and test set. This indicates that the model learns patterns that are specific to the encryption under a fixed key. This would hamper the use of the model in practice: During deployment, the key by which an encrypted message was produced will be unknown to the operator.

We find, however, that a model trained on a dataset in which ciphertexts have been generated under a frequently rotating key, can compensate for this effect. Fig. 3 *Right* shows that the model trained on $\mathtt{M}^{\mathtt{train}}_{\mathtt{rot}}$ has a lower false negative rate on both test sets. The overall accuracy of this model reaches up to 99.96%.

Can the model correctly classify ciphers it has not seen before? In the previous experiments, the model's training set included the full set of ciphers that were used to encrypt messages in the training set. Fig. 4 shows the performance of two models trained on $\mathtt{M}^{\mathtt{train}}_{\mathtt{fixed}}$ and $\mathtt{M}^{\mathtt{train}}_{\mathtt{rot}}$, respectively, where one of the ciphers was excluded from the training set and the

[1]See notebook Baseline Comparison provided in repository for details [8]

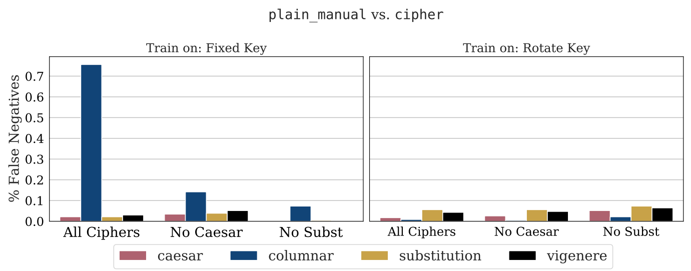

Fig. 4. *Left*: False negative rate of a model trained on $\mathtt{M}^{\mathtt{train}}_{\mathtt{fixed}}$ with different subsets of ciphers included in the train set *Right*: False negative rate on the same test sets for a model trained on $\mathtt{M}^{\mathtt{train}}_{\mathtt{rot}}$.

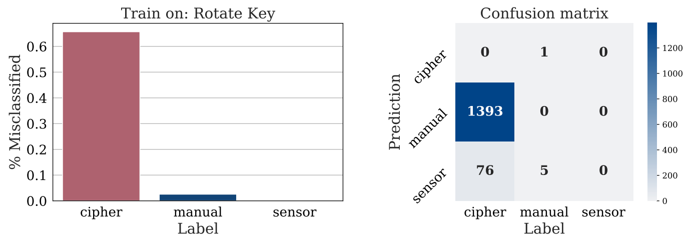

Fig. 5. *Left*: Error rate across classes *Right*: Each square indicates the number of records from a class shown on the x-axis wrongly labelled as the class shown on the y-axis.

model observed examples of this type during the test phase only. This does not seem to impact the performance of either model. Neither excluding caesar nor substitution ciphers from the training set significantly increases the false negative rate for these classes in the test set.

Can the model distinguish plaintext messages containing sensor data from weak substitution ciphers? We further assessed the accuracy of a model trained to distinguish messages labelled as `plain_sensor` from those generated by a set of weak substitution ciphers. We expected this to be a harder task for the model as the data generated by an aircraft's internal sensor systems has a higher entropy and closely resembles the output of some ciphers. We find, however, that the model has an equally high success rate (up to 99.96% accuracy) in distinguishing between these two classes when trained on a diverse set of ciphers. Only if the model has been trained on fixed key ciphers that differ from the test set, accuracy slightly degrades (false negative rate of 1.8% for a model trained on $\mathtt{S}^{\mathtt{train}}_{\mathtt{fixed}}$ evaluated on $\mathtt{S}^{\mathtt{test}}_{\mathtt{diff}}$).

Can the model perform multi-class classification? Last, we assessed whether the model can distinguish between more than two classes and correctly classify messages as either `plain_manual`, `plain_sensor`, or `cipher`. Fig. 5 shows that the model can distinguish between these message content types with high accuracy: With an overall accuracy of 99.49% the model clearly outperforms the baseline model with 80.33%. We find that the model is most likely to misclassify messages encrypted with weak substitution ciphers (label `cipher`) as `plain_manual` and has the lowest error rate on records of type `plain_sensor` (see Fig. 5 *Right*).

VI. Discussion

These results are highly encouraging. A model trained on a sufficiently diverse set of ciphers is able to distinguish between ciphertext and plaintext messages with high accuracy and significantly outperforms previous methods[2].

It remains to be seen, however, whether results from the pre-processed, cleaned data presented here are generalisable to a wider dataset. In particular, the model was trained and tested on messages submitted via satellite only. Messages on other channels might be subject to a distributional shift and hence lead to a decline in model performance. Furthermore, only messages with particular messages tags (see Section II) were taken into account. Other classes might contain harder to classify message contents.

If the approach presented here proves to be robust to the conditions described above, a next step will be to explore whether it is possible to also distinguish insecure ciphers from standardised, secure encryption. This would further enhance the utility of the model in practice.

Acknowledgments

We would like to thank Kasra EdalatNejad for his supervision and the time and effort he put into reading long technical specification documents to understand and clean the dataset provided.

References

[1] Aeronautical Radio Inc., "ARINC Communication Addressing & Reporting System," https://web.archive.org/web/20110706055818/http://www.arinc.com/downloads/product_collateral/acars_first_datasheet.pdf, 1978.

[2] M. Smith, D. Moser, M. Strohmeier, V. Lenders, and I. Martinovic, "Undermining privacy in the aircraft communications addressing and reporting system (acars)," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 3, pp. 105–122, 2018.

[3] Federal Aviation Administration, "Access to aircraft situation display (asdi) and national airspace system status infor- mation (nassi)," https://www.federalregister.gov/documents/2011/03/04/2011-4955/access-to-aircraft-situation-display-asdi-and-national-airspace-system-status-information-nassi, 2011.

[4] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*. Springer, 1998, pp. 137–142.

[5] S. Singh, *The code book*. Doubleday New York, 1999, vol. 7.

[6] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, pp. 3111–3119, 2013.

[7] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[8] T. Stadler, F. Intoci, and M. Mariantoni, "Automatic detection of weak cipher usage in aricraft communications," https://github.com/CS-433/cs-433-project-2-acars, 2020.

[9] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS workshop*, no. CONF, 2011.

[2]Some previous research exists but can not be directly presented here due to non-disclosure agreements