Unmasking Concealed 5G Privacy Identity with Machine Learning and GPU in 12 mins

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY-NC-SA 4.0

SUBMISSION DATE / POSTED DATE

04-11-2020 / 04-11-2020

CITATION

Tea, Vui Huang (2020): Unmasking Concealed 5G Privacy Identity with Machine Learning and GPU in 12 mins. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.13187636.v1>

DOI

[10.36227/techrxiv.13187636.v1](https://doi.org/10.36227/techrxiv.13187636.v1)

Unmasking Concealed 5G Privacy Identity with Machine Learning and GPU in 12 mins

Tea Vui Huang

Abstract The 3rd Generation Partnership Project (3GPP) standard for 5G telecommunications specifies privacy protection schemes to cryptographically encrypt and conceal permanent identifiers of subscribers to prevent them from being exposed and tracked by over-the-air eavesdroppers. However, conventional privacy-preserving protocols and architectures alone are insufficient to protect subscriber privacy as they are vulnerable to new types of attacks due to the utilization of the emerging technologies such artificial intelligence (AI). A conventional brute force attack to unmask concealed 5G identity using a CPU would require ~877 million years. This paper presents an apparatus using machine learning (ML) and a graphics processing unit (GPU) that is able to unmask a concealed 5G identity in ~12 minutes with an untrained neural-network, or ~0.015 milliseconds with a pre-trained neural-network. The 5G concealed identities are effectively identified without requiring decryption, hence severely diminishing the level of privacy-preservation. Finally, several ML defense countermeasures are proposed to reestablish privacy protection in 5G identity.

Index Terms— 5g, artificial intelligence, classification, gpu, machine learning, mobile network, neural networks, phones, privacy

I. INTRODUCTION

According to the system architecture for the 5G System [1], a globally unique 5G SUPI (Subscription Permanent Identifier) [4] shall be allocated to each subscriber in the 5G System. When a 5G user equipment (UE) needs to indicate its SUPI to the 5G Radio Access Network (NG-RAN) (e.g. as part of the registration procedure), the UE provides the SUPI in concealed form known as the SUCI (Subscription Concealed Identifier). Concealment enables identity privacy and makes the subscriber's geographical presence and mobile connections untraceable to over-the-air eavesdroppers. The un-linkable temporary ephemeral privacy preserving SUCI makes it infeasible for it to be connected together to deduce the subscriber's permanent identity. Revealing a never-changing permanent identity would help observers to track the user. An IMSI (International Mobile Subscriber Identifier) based SUPI may further help the tracking as it may be used in other contexts [3].

In operation, the UE conceals the permanent SUPI identity in a temporal SUCI identity and transmits it over-the-air via 3GPP RAT (Radio Access Technology) such as 5G NR, and non-3GPP RAT such as Wi-Fi [7]. The base

station and access points relay the SUCI to the 5GC Core (5GC), where the SIDF (Subscription Identifier De-concealing Function) de-conceals the SUCI back to the original SUPI. It has been shown that a quantum adversary capable of issuing quantum queries to an appropriate quantum computer can easily break ECIES-based identification scheme [5] by employing Shor's quantum algorithm [6]. This paper demonstrates how 5G subscriber privacy and anonymity can be unmasked on non-quantum, conventional computers with GPU by using deep learning ML classification methods to link temporary ephemeral ciphertexts without necessitating the de-concealment of the SUPI permanent identifier.

A. Plaintext Scheme Input

A valid SUPI can be either an IMSI for 3GPP RAT, or NAI (Network Access Identifier) for non-3GPP RAT [7]. IMSI is composed of MCC, MNC and MSIN. NAI for SUPI has the form "username@realm" [8]. For IMSI-format SUPI, only the MSIN part is concealed. For NAI-format SUPI, only the username part of the NAI is concealed. The IMSI home network identifier and NAI realm are transmitted in plaintext. For example: If the SUPI IMSI is "274120999999999", where MCC is "274", MNC is "12" and MSIN is "0999999999", then the SUCI ciphertext is the encryption of "0999999999". If the SUPI NAI is "user17@example.com", where username is

“user17” and realm is “example.com”, then the SUCI ciphertext is the encryption of “user17”. The example above is computationally equivalent to a NAI-based SUPI which has a NAI-username of IMSI-MSIN format, i.e. a NAI of “0999999999@example.com” where username is “0999999999” and realm is “example.com”.

B. Ciphertext Scheme Output

According to the 5G specification, it is mandatory for the UE and network to implement the SUPI protection schemes ECIES Profile A and ECIES Profile B [2], which are considered secure against a classical adversary [5]. Both profiles calculate and generate the SUCI scheme output in the same manner. The UE generates a random elliptic curve ephemeral key pair and uses the ephemeral private key with the home network public key to create an ephemeral shared key via an elliptic curve key exchange. An ANSI-X9.63-KDF (Key Derivation Function) [9] is then used to derive the ephemeral encryption key, ICB (Initial Counter Block) and the ephemeral MAC (Message Authentication Code) signing key. The ephemeral encryption key and the ICB is used to encrypt the plaintext block (the nibble-swapped IMSI MSIN, or NAI username) using the AES-128 encryption algorithm in CTR counter mode [16] to create the cipher-text. The ephemeral MAC key signs the ciphertext to using HMAC-SHA-256 create the MAC. The frequency the home network generates new public/private key pair and how the public key is provisioned to the UE are out of the scope of the 5G specification [2]. Aside from ECIES Profile A and B, the HPLMN (Home Public Land Mobile Network) can use null-scheme (unconcealed SUPI) and proprietary protection schemes. Alternative protection schemes are also available in various research papers [5].

C. Theory

A SUCI is composed of SUPI type, home network identifier, routing indicator, routing indicator, protection scheme identifier, home network public key identifier and protection scheme output (ciphertext). The ciphertext is encrypted with the symmetric ephemeral encryption key and the ephemeral ICB. In terms of life-spans, the scheme input is perpetual, the home network public and private keys can be perpetual, long-term or medium-term, and everything else is ephemeral. Examining the SUCI generation elements and their life-span, it can be observed that the plaintext scheme input containing the SUPI unique identifier is constant, and everything else is transient noise that engulfs it to create the SUCI ciphertext value. Consequently, the noisy SUCI ciphertext could be classified using deep learning to link it back to the constant scheme input that was used to create it. Classification is a supervised deep learning method that requires labelled training data. Hence if we extract the SUCI ciphertext as the feature and the scheme input (5G identity) as the label, we can create a labelled training and test data that can be used to train a deep learning classification model to predict if a ciphertext conceals a 5G identity of a subscriber,

without having to decrypt the SUCI.

II. METHODS AND PROCEDURES

Classification is a deep-learning method that uses pattern learned from trained data to determine the category of an item in a given data. We will use a binary classification to predict whether a SUCI conceals a 5G identity, hence the first task is to compose data suitable for training such a model. There are two sets of data required to train the model. These are the training data used for training the model weights and biases, and test data for evaluating the model on unseen ciphertext. These data are synthetically generated offline without any local nor remote access to the 5G device nor any over-the-air data. Data preparation has to be performed on the ciphertext to enable machine learning algorithms to uncover insights and make predictions. The assumption is that either the IMSI or the NAI of the target 5G identity is known, and the home network public key is known or has somehow been acquired.

A. Generation of Training and Test Data

The data is prepared for the classifier by creating SUCI that belong to the targeted IMSI home network or NAI realm, and generated with the same SUPI protection scheme and home network public key. The training and test data consist of two groups, one conceals the target identity and the other conceals non-target identities. To ensure that the data is balanced for the binary classifier, the training dataset should be a 50:50 ratio of both groups. To create the former group, fresh subscriber ephemeral keys is generated and used with the parameters above and a constant IMSI-MSIN or NAI-username as input to the SUCI algorithm [2]. To create the latter group, fresh subscriber ephemeral keys is generated and used with the parameters above and a random or consecutive sequence of IMSI-MSIN or NAI-username as input to the SUCI algorithm [2].

B. Feature Engineering

Feature engineering involves rearranging, combining, and otherwise manipulating data to derive new feature variables that would improve the model’s prediction.

Feature Selection and Data Transformation

In feature selection, we identify the input that has the biggest influence on predicting the outcome to generate the best training results. In protection scheme ECIES Profile A, the SUCI consist of unconcealed headers followed by a 256-bit elliptic curve public key, 40-bit protection scheme output (ciphertext) and 64-bit MAC. E.g. “suci-0-274-12-678-1-28-7956 ... b69e9a186b ... c2c24e3cc56324f8” where the ciphertext is “b69e9a186b”. As the constant that will uniquely identify the subscriber is concealed only in the ciphertext, we select the ciphertext as the feature and discard the rest of the data. We use the scheme input as the label and transform it by replacing all positive labels with “1” and all negative labels with “0”.

	Unconcealed 5G Identifier	Train & Test Labels	
		Does not conceal 00012080f6 (No)	Conceals 00012080f6 (Yes)
1	00012080f6	0.0	1.0
2	00919272f9	1.0	0.0
3	00016633f6	1.0	0.0
4	00012080f6	0.0	1.0
5	00012080f6	0.0	1.0

Table. 1. Examples of the 2 labels derived from the ciphertext. In the ‘Yes’ label column, entries that conceal the 5G identifier to be classified are marked with ‘1.0’ and all other entries that do not match are marked with ‘0’. This is performed similarly for the ‘No’ label column.

Separation of Features and Sorting by Labels

The 10 hexadecimal characters of the ciphertext are separated equally into 5 feature columns, each consisting of 2 hexadecimal characters ranging from ‘00’ to ‘FF’ and maintain the same sequence as the original ciphertext. These features are labelled c1, c2, c3, c4 and c5.

	Unconcealed 5G Identifier	Concealed SUCI Ciphertext	Train & Test Features for Neural-Network Model				
			c1	c2	c3	c4	c5
1	00012080f6	85c0fcc30	85	c0	fc	ca	30
2	00919272f9	ed4bc3ca50	ed	4b	c3	ca	50
3	00016633f6	6fe2e11ea1	6f	e2	e1	1e	a1
4	00012080f6	50fec5d46a	50	fe	c5	d4	6a
5	00012080f6	2ccd664e38	2c	cd	66	4e	38

Table. 2. Examples of the 5 features derived from the ciphertext.

Sequencing of Features

Although the ciphertext rows are now grouped, the connection between the features are not yet apparent. To enhance this, we apply n-gram sequencing to the data. An n-gram is a contiguous sequence of n items from a given sample of text. It is widely used in statistical NLP (Natural Language Processing) but can be used for other types of data. For example, it has been successfully used in the identification species from short sequences of DNA [11].

	Train & Test Features				Labels	
	c1c2	c2c3	c3c4	c4c5	No	Yes
1	85c0	c0fc	fcc3	ca30	0.0	1.0
2	ed4b	4bc3	c3ca	ca50	1.0	0.0
3	6fe2	e2e1	e11e	1ea1	1.0	0.0
4	50fe	fec5	c5d4	d46a	0.0	1.0
5	2ccd	cd66	664e	4e38	0.0	1.0

Table. 3. Examples of the 4 n-gram features and 2 labels combined.

Factorization of Features

To simplify connections of the n-grams, we factorize the ciphertext features as categorical variables in each individual feature column. Factorization obtains a simpler, sequential numeric representation of the ciphertext features to identify the distinct values.

The combined feature and label data are first sorted in descending order by the ‘Yes’ label. This results in data sets

concealing the target identifier at the top, and non-concealing data sets at the bottom.

The number of unique values in each feature column may not be the maximum possible unique values. If this occurs, some ciphertext cannot be enumerated to the factorized values when the model is asked to do classification prediction after it is trained. To resolve that, dummy data entries are appended to ensure each feature column has the maximum possible unique values. These dummy data values are removed after factorization and are excluded from the train and test set.

On factorization of each feature column, the first distinct value is replaced with the value 0, and the second distinct value is replaced with the value 1, etc. From casual observation, it can be readily guessed that low value sets in the top rows has a higher probability of concealing the target identifier than those at the bottom rows.

	Train & Test Features				Labels	
	c1c2	c2c3	c3c4	c4c5	No	Yes
1	0	0	0	0	0.0	1.0
2	1	1	1	1	0.0	1.0
3	2	2	1	2	0.0	1.0
4	3	1	2	3	1.0	0.0
5	4	3	2	3	1.0	0.0

Table. 4. Examples of the train and test data with 4 factorized features and 2 labels.

Feature Scaling

Finally, we apply min-max normalization feature scaling to bring all the feature values into the range of 0 and 1. For each factorized column the minimum value is always 0, hence each feature column is divided by the maximum value of 65535 as the range for each factorized feature column is the hex value ‘0000’ to ‘FFFF’.

To create a classifier that can make accurate prediction on unseen ciphertext, part of the available data is reserved as a test set for evaluating the classifier, while the rest is used for training, e.g. 60% for training and 40% for testing. The data can be shuffled before splitting to obtain different accuracy and loss results.

	Train Features				Labels	
	c1c2	c2c3	c3c4	c4c5	No	Yes
1	0.002029	0.460770	0.460892	0.459763	0.0	1.0
2	0.671036	0.479935	0.635437	0.668533	1.0	0.0
3	0.551178	0.750717	0.503281	0.700577	1.0	0.0
4	0.352661	0.352737	0.094788	0.336044	0.0	1.0
5	0.179947	0.180573	0.180008	0.140320	0.0	1.0

	Test Features				Labels	
	c1c2	c2c3	c3c4	c4c5	No	Yes
1	0.071045	0.685776	0.686859	0.518845	1.0	0.0
2	0.555786	0.706543	0.608017	0.704056	1.0	0.0
3	0.261520	0.261612	0.261337	0.245544	0.0	1.0
4	0.372757	0.410446	0.610367	0.132843	1.0	0.0
5	0.241364	0.587997	0.071671	0.096725	1.0	0.0

Table. 5. Examples of the resulting train and test data containing the 4 features and 2 labels. The feature data has been sorted, n-gram, factorized and normalized; while the label data is grouped as 'No' if the feature set represents a ciphertext that does not conceal the specified 5G identifier and 'Yes' otherwise. Note: The data is not related to other examples.

C. Neural Network Model Training

The deep learning software Keras & TensorFlow is used to create the neural network model. It is configured with losses and metrics, trained with the data and finally used to perform the ciphertext classification. TensorFlow supports the use of GPU and was developed by Google to conduct machine learning research. Keras is the high-level programming interface of TensorFlow, its core data structures are layers and models.

Keras Layers	Type	Output Shape	Param #
Layer 1 (relu)	Dense	(None, 4)	20
Layer 2 (relu)	Dense	(None, 7)	35
Layer 3 (softmax)	Dense	(None, 2)	16
Trainable parameters			71

Table. 6. Keras neural-network model accepting 4 feature inputs derived from the 5G SUCI ciphertext, and generates 2 classification outputs predicting if the ciphertext conceals a specified 5G identifier or not.

The Keras sequential model for the classifier contains a linear stack of 3 dense layers. The layers 1, 2 and 3 has output space dimensionality of 4, 7 and 2. The layers use the activation functions rectified linear unit (relu), relu and softmax respectively to define their output. Softmax is used for the last layer as its result can be interpreted as a probability distribution. The model's training configuration is an Adam optimizer, a categorical cross-entropy loss function and an accuracy metric. Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments [12]. Categorical cross-entropy loss function computes the loss that a model should seek to minimize between the labels and predictions. Accuracy metric calculates how often predictions equals labels, and this will be evaluated by the model during training and testing. The model is trained for 100 epochs, where epoch is an iteration over the data. A fixed epoch can result in underfitting and overfitting, so it may need to be fine-tuned. Model accuracy and loss is obtained at each epoch to monitor training progress.

The GPU used for the deep-learning computation is the "NVIDIA GP108 GPU". Hardware-wise, it has a base clock of 1519 MHz, features 384 shading units and uses GDDR5 memory with a 64-bit interface. Software-wise, it supports Compute Unified Device Architecture (CUDA) parallel computing platform, with a Compute Capability of 6.1. In terms of performance based on architecture, shader count and clocks, the newer "NVIDIA GeForce RTX 3090" launched in September 2020 is 14x faster.

Fig. 1. GPU load for neural-network training of 100 epochs in 10 mins.

Computing a 100,000 training & testing data set takes ~2.5 minutes on a 1.80 GHz Intel Core i7-10510U CPU & model training of 100 epochs takes ~10 minutes on a 1.52 GHz NVIDIA GP108 GPU. As such, it takes approximately 12.5 minutes from data set creation to completion of model training for unmasking each 5G identity. This sequence can be pre-computed for faster deployment. To do that, the trained model can be saved into a Hierarchical Data Format (HDF) file containing the model's architecture, weights values, and computation information. It does not require the model building code to run, making it available for sharing or deploying. The total file size to unmask one 5G identity is ~720 Kbytes, of which ~30 Kbytes is for the model and ~690 Kbytes for the enumeration file.

Fig. 2. GPU load for neural-network making 1 mil predictions in 15 sec.

	Unconcealed 5G Identifier	Concealed SUCI Ciphertext	Conceal No (%)	Conceal Yes (%)	
1	00012080f6	657966678d	2.1471	97.8529	OK
2	00012080f6	7e5637d590	2.1503	97.8497	OK
3	00012080f6	2ccd664e38	2.2289	97.7711	OK
4	00012080f6	b526f03b56	2.8379	97.1621	OK
5	00012080f6	6fe2e11ea1	2.8900	97.1100	OK
6	00919272f9	50fec5d46a	7.8654	92.1346	-
7	00016633f6	78167bd4bb	98.7021	1.2979	OK
8	00011746f9	85c0fcca30	100.0000	0.0000	OK
9	00319625f4	545e9b016f	100.0000	0.0000	OK

10	00715752f3	0435c7701d	100.0000	0.0000	OK
11	00818158f7	ed4bc3ca50	100.0000	0.0000	OK
12	00714233f7	ff81209d84	100.0000	0.0000	OK

Table 7. Example prediction results for a neural-network model trained to classify if a SUCI ciphertext conceals the 5G identifier ‘00012080f6’. In cases #1-5, it predicted correctly that the ciphertext conceals the 5G identifier. In cases #7-12, it predicted correctly that the ciphertext does not conceal the 5G identifier. In case #6, it predicted wrongly that the ciphertext conceals the 5G identifier.

A conventional brute force attack to unmask the concealed 5G identity using a CPU would require ~ 877 million years. Using ML & GPU, the 5G identity can be unmasked in ~ 12 minutes with an untrained neural-network, or ~ 0.015 milliseconds with a pre-trained neural-network.

III. RESULTS

The deep learning classifier described in this paper is applied on 100,000 5G SUCI from the public data “5G Subscription Concealed Identifier (SUCI) Of IMSI-Based Subscription Permanent Identifier (SUPI), Privacy Protected with ECIES Profile A Protection Scheme” [13] that is modelled after the 5G implementer’s test data [3]. Only the SUCI containing the ciphertext scheme output and the plaintext containing the scheme input is used from the data. The data corresponds to SUCI computation for IMSI-based SUPI and SUPI protection scheme ECIES Profile A. Each SUCI contains a 256-bit ephemeral public key, 64-bit MAC & 40-bit encrypted MSIN. The SUCI are generated using a single home network key set. It is a balanced dataset with 50% of the SUCI concealing one 5G identity, and 50% concealing other identities. The classifier is trained on 60,000 samples and tested on 40,000 samples. In a sample run, the model is able to obtain a high validation accuracy of 95.48% and low validation loss of 0.13.

A. Model Accuracy and Loss

Accuracy measures the percentage of matching predictions made by the model as compared to actual value of the label data, i.e. if the ciphertext conceals the 5G identity. Lower losses indicate better model performance. The accuracy and loss are measured on both training and test dataset after each training epoch. In Fig. 3, the model starts from a high accuracy and improves very quickly along a steep slope in less than 10 epochs. The test accuracy mirrors the training accuracy. In Fig. 4, the model loss decreases rapidly along the steep slope in 10 epochs. The test loss similarly mirrors the training loss.

Fig. 3. Neural network model training and test accuracy vs epoch. The test accuracy mirrors the training accuracy. The model learns very quickly from the feature engineered data and has high accuracy.

Fig. 4. Neural network model training and test loss vs epoch. The test loss mirrors the training loss. The model learns very quickly from the feature engineered data and has low loss.

B. Area Under the Receiver Operating Characteristic

The area under the receiver operating characteristic (AUROC) is a performance metric used to evaluate classification models. It indicates the probability that the ciphertext will be correctly predicted to be concealing a 5G identity. The closer the curve is to the vertical TPR axis, the more accurate the classifier is. A classifier with AUROC of 0.5 has a 50% chance of making correct predictions, whereas one with an AUROC of 1.0 is considered a perfect classifier [14]. In Fig. 5, the classifier has an AUROC of 0.98, making it a near-perfect classifier.

Fig. 5. Neural network model Area Under the Receiver Operating Characteristic (AUROC). It has an AUROC of 0.98, making it a near-perfect classifier.

C. Confusion Matrix

A confusion matrix summarizes the performance of a classification model on a set of known test data. True Positive (TP) is the number of ciphertext that is correctly classified to be concealing the 5G identity, while False Positive (FP) is the opposite. True Negative (TN) is the number of ciphertext that is correctly classified to be not concealing the identity, while False Negative (FN) are the number that are incorrect. Positive Precision measures the ability of the classifier not to label as positive a negative ciphertext, and Positive Recall measures the ability of the classifier to find all positive ciphertexts. F1 measures the balance of precision and recall, the best score is 1.

In Table. 8, TP is observed to be roughly equal to TN, indicating that the ciphertext is correctly classified to be concealing the 5G identity as often as it is correctly classified to be concealing other identity. FP is larger than FN, indicating that the ciphertext has higher probability of being incorrectly classified to be concealing the 5G identity. The accuracy is a high 95.48%. The classifier's high positive recall is greater than its positive precision, showing that it has a good ability to identify ciphertexts that are concealing the 5G identity. The classifier has high F1 scores, indicating a good balance of precision and recall.

Confusion matrix		
n=40,000 test samples	Score positive	Score negative
Actual positive	19,923 TP	125 FN
Actual negative	1,684 FP	18,268 TN
Accuracy	95.48%	
Misclassification	4.52%	
	Positive	Negative
Number of cases	20,048	19,952
Precision	0.92	0.99
Recall	0.99	0.92
F1	0.96	0.95

Table. 8. Confusion matrix showing good performance of the classifier.

IV. DISCUSSION

A key risk assessment process is to identify functions required to minimize known risks to an acceptable level. The privacy protection comes from the fact the SUPI is concealed and un-linkable for the eavesdroppers to deduce the subscriber's SUPI. This paper has conversely shown that the concealed SUCI identity can be accurately linked back to the unconcealed permanent SUPI identity using artificial intelligence without having to de-conceal the SUPI identity. The over-the-air eavesdroppers can monitor and capture all SUCI at any given point in time and at multiple locations. As only one SUCI concealing a unique SUPI can be sent at any point in time, it is trivial to determine when and where a subscriber is by using the classifier. As SUCI is a 5G standard, the vulnerability affects all 5G compliant equipment that handles SUCI, including the USIM, eSIM, iSIM, mobile phones, IoT devices and base stations.

In this section, we present various countermeasures that can be taken by the subscribers and the home network to thwart and reduce the effectiveness of this artificial intelligence approach. The objective of classifier-resistant SUPI protection schemes is to make it harder for the classifier to be trained, and/or to nullify its result. This is akin to creating camouflage to fool vision algorithms by complicating the decisive facial features the AI relies on to recognize a face [15]. Proprietary protection schemes can be used to implement classifier-resistant countermeasures that are not 5G compliant. These require end-to-end synchronization and care should be taken when using proprietary schemes for small groups of users [2].

A. Increase Scheme Input Length

The classifier relies on n-gram sequencing of the ciphertext to make classification predictions. For IMSI-based SUCI operating on 3GPP RAT, the 10-character IMSI-MSIN plaintext scheme input requires only 4 n-gram features to train the classifier. The same applies to 10-character NAI-username plaintext scheme input on non-3GPP RAT. To increase classifier training difficulty, one solution is to increase the scheme input length, which increases the number of n-gram features required. Increasing n-gram features increases the permutations of factorized features. This rapidly increases the number of samples required as well as the number of model parameters that has to be trained.

For the example in Fig. 7 and Fig. 8, the scheme input is lengthened from the original 40-bit to 168-bit. The lengthened scheme input consists of a proprietary 128-bit ephemeral symmetric key concatenated with a 40-bit ciphertext that is the AES-CTR encryption of the original 40-bit scheme input using the ephemeral symmetric key and a fixed ICB of zero values. It is assumed that the ephemeral symmetric key cannot be recovered from the scheme output. Protection scheme ECIES Profile A is then applied

to the lengthened scheme input to generate the concealed scheme output. The classifier will now require 20 n-gram features to train. The net effect as compared to Fig. 3 and Fig. 4 is that the gentler slopes show that the model learns the ciphertext pattern slower and takes more epochs to increase accuracy and reduce loss.

Fig. 6. GPU load for neural-network training, 100 epochs in 10 mins. There is a heavier load on the GPU as compared to Fig. 1

Fig. 7. Neural network model training and test accuracy vs epoch. Test accuracy mirrors training accuracy. The model learns slower and takes more epoch to increase accuracy with longer scheme input.

Fig. 8. Neural network model training and test loss vs epoch. Test loss mirrors training loss. The model learns slower and takes more epoch to reduce loss with longer scheme input.

The challenge for long NAI-username is for the subscriber to memorize and use such a long username, and for the non-3GPP RAT equipment to support these long usernames. As for IMSI-MSIN, the length is fixed in the 5G specification [2] and many non-standard changes need to be made across the 5G system to support this modification. However, IMSI-MSIN scheme input length can be accommodated if a new format is defined in a proprietary USIM-base SUPI protection scheme.

B. Noisy Scheme Input

Another way to hinder the classifier is to increase the variation of the plaintext scheme input indeterminately such that it becomes difficult for the neural network model to learn the pattern in the dataset. One method is for the UE to randomly generate a fresh ephemeral symmetric key and counter-block and use them to encrypt the static scheme input before applying the protection scheme. The SIDF must have the same key and counters to decrypt the ciphertext, so the ephemeral symmetric key and counter is concatenated or otherwise included with the ephemerally encrypted scheme input before they are concealed. The added noise causes the model to learn the noise and deviate from discovering the actual ciphertext pattern.

For the example in Fig. 10 and Fig. 11, the scheme input is lengthened with noise from the original 40-bit to 296-bit. The input consists of a proprietary 128-bit ephemeral symmetric key concatenated with a proprietary 128-bit ephemeral ICB and a 40-bit ciphertext that is the AES-CTR encryption of the original 40-bit scheme input using the ephemeral symmetric key and ephemeral ICB. It is assumed that the ephemeral symmetric key and ICB cannot be recovered from the scheme output. ECIES Profile A is applied to the scheme input to generate the concealed scheme output. The model will now require 36 n-gram features to train. The net effect as compared to Fig. 3 and Fig. 4 is that the noisy scheme input diverges the train and test curves and the model is not able to generalize the data. Generalization refers to the model's ability to adapt properly to new, previously unseen data, drawn from the same distribution as the one used to create the model. As a result, the model trained with noisy scheme input has lower accuracy and higher losses when classifying ciphertext that it has not seen before.

Fig. 9. GPU load for neural-network training, 100 epochs in 10 mins.

Fig. 10. Neural network model training and test accuracy vs epoch. Test accuracy is lower than training accuracy. The model trained with noisy scheme input has lower accuracy when classifying ciphertext that it has not seen before. Note the disparity between train and test curves.

Fig. 11. Neural network model training and test loss vs epoch. Test loss is higher than training loss. The model trained with noisy scheme input has higher losses when classifying ciphertext that it has not seen before. Note the disparity between train and test curves.

C. Noise Generation from Base Stations

A key adversary of neural network models is noisy data. Noise is meaningless data that has no correlation to the data pattern and misguides the model. If not removed or addressed, noise can cause the model to incorrectly learn

from it and eventually dilute the model's prediction and classification accuracy, and increase its errors and losses. Hence one effective countermeasure against classifiers is to flood it with noise. SUCI is collected when it is sent from the UE to the base stations (BS). The redress is for the base stations to generate noise by masquerading as the UEs and sending simulated SUCI of real and fake SUPI to themselves. As the generator of the noise, the BS will be able to filter out these simulated SUCI connections and prevent them from being relayed to the SIDF for de-concealment. The BS can broadcast simulated SUCI connections at various hours and frequency to smokescreen the location and movement of the 5G subscriber from the eavesdroppers. The subscriber will appear to the classifier to be present at multiple locations at the same time, or all the same location all the time. These simulated transmissions can also be made at different power levels to simulate the UE connecting from various distances. The effect is a reverse adversarial attack carried out by the home network to fool the eavesdropper's ML models. The complication is the increased processing power requirement on the BS, radio interference and radio traffic generated.

V. CONCLUSION

In this paper we described how the identity privacy in the current 5G specification of March 2020 [2] can be unmasked in a short time on a GPU-powered deep-learning neural network with 95% accuracy. Volumes of training and test data can be created without requiring any physical or remote access to the 5G equipment.

We discussed potential defensive classifier-resistant countermeasures. For an privacy protection scheme to be resilience against ML, the suggestion is that the scheme input should be a long, non-constant noisy value. In particular, the ciphertext should not be the result of a single symmetric encryption of a constant value. The suggestion is for the design and specification of any future privacy protection scheme to be tested against the deep learning classifier described in this paper as an ML/AI vulnerability detection to verify that the proposed schemes are not susceptible to deep learning classification.

REFERENCES

- [1] 3rd Generation Partnership Project: 5G; *System architecture for the 5G System (5GS) (3GPP TS 23.501 version 15.9.0 Release 15)* (Mar 2020), <https://www.3gpp.org/DynaReport/23501.htm>
- [2] 3rd Generation Partnership Project: *Security architecture and procedures for 5G System (3GPP TS 33.501 version 15.8.0)* (Mar 2020), <https://www.3gpp.org/DynaReport/33501.htm>
- [3] Internet Engineering Task Force, *RFC 4187 Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)*, <https://tools.ietf.org/html/rfc4187>
- [4] Shoup, V.: *A proposal for an ISO standard for public key encryption*. IACR Cryptology ePrint Archive 2001, 112 (2001), <http://eprint.iacr.org/2001/112>
- [5] Khan H., Dowling B., Martin K.M. (2018), *Identity Confidentiality in 5G Mobile Telephony Systems*. <https://helda.helsinki.fi/handle/10138/308155>
- [6] Shor, P.W.: *Algorithms for quantum computation: Discrete logarithms and factoring*. In: 35th Annual Symposium on

- Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994. pp. 124134. IEEE Computer Society (1994). <https://doi.org/10.1109/SFCS.1994.365700> SFCS. 1994.365700
- [7] 3rd Generation Partnership Project: *Policy and charging control framework for the 5G System (5GS); Stage 2 (3GPP TS 23.503 version V16.5.0) (Jul 2020)*, <https://www.3gpp.org/DynaReport/23503.htm>
 - [8] Internet Engineering Task Force, *RFC 7542 The Network Access Identifier*, <https://tools.ietf.org/html/rfc7542>
 - [9] SECG SEC 1: *Recommended Elliptic Curve Cryptography, Version 2.0, 2009*. <http://www.secg.org/sec1-v2.pdf>
 - [10] 3rd Generation Partnership Project: *Common Data Types for Service Based Interfaces, (3GPP TS 29.571 version 16.3.0) (Jul 2020)*, <https://www.3gpp.org/DynaReport/29571.htm>
 - [11] Tomović, Andrija; Janičić, Predrag; Kešeljić, Vlado (2006). *n-Gram-based classification and unsupervised hierarchical clustering of genome sequences*. Computer Methods and Programs in Biomedicine. 81 (2): 137-153. <https://doi.org/10.1016/j.cmpb.2005.11.007>
 - [12] Diederik P. Kingma, Jimmy Ba. Adam: *A Method for Stochastic Optimization* (2014). arXiv:1412.6980, <https://arxiv.org/abs/1412.6980>
 - [13] Tea Vui Huang (2020). *5G Subscription Concealed Identifier (SUCI) Of IMSI-Based Subscription Permanent Identifier (SUPI), Privacy Protected with ECIES Profile A Protection Scheme*, IEEE Dataport. <https://doi.org/10.21227/nmae-8y91>
 - [14] Fawcett, Tom (2006). *An Introduction to ROC Analysis*. <https://people.inf.elte.hu/kiss/11dwhdm/roc.pdf>
 - [15] Feng, Ranran; Prabhakaran, Balakrishnan (2013). *Facilitating Fashion Camouflage Art*. Proceedings of the 21st ACM International Conference on Multimedia. MM '13. ACM: 793–02. doi:10.1145/2502081.2502121. ISBN 978-1-4503-2404-5.
 - [16] Internet Engineering Task Force, *RFC 3686 Using Advanced Encryption Standard (AES) Counter Mode with IPsec Encapsulating Security Payload (ESP)*, <https://tools.ietf.org/html/rfc3686>

Tea Vui Huang is the inventor of over 40 ML/AI, 4G/5G, IoT and cybersecurity patents that have been cited by world-leading cybersecurity vendors such as Idemia Identity & Security, Thales Eecom, Broadcom Symantec, Dell EMC, Kudelski NagraVision as well as Fortune 500 companies such as Google, Samsung, Qualcomm, IBM, HP, Visa & Tencent.