https://doi.org/10.2352/ISSN.2470-1173.2021.4.MWSF-344

# AiroIdent – User identification based on analyzing WPA2 encrypted traffic containing search engine interactions

***Mario Hildebrandt, Aamir Shakir, Alexander Ziemke, Mohamed Abdelrazek, Hannes Stuetzer, Dominik Blut, Kevin Lamshoeft, Salatiel Ezennaya-Gomez, Christian Kraetzer, Jana Dittmann; Research Group Multimedia and Security, Otto-von-Guericke-University; Magdeburg, Germany***

## Abstract

*Most search engines provide search suggestions and autocompletion mechanisms based on the partially typed search string. In order to implement such functionality, frequent requests are being sent to the search engine provider. Recent publications show that there is a risk that the user can be identified by observing the TLS encrypted traffic and analyzing the unencrypted meta data. In this paper we extend this approach to the observation of widely used encrypted WiFi networks in order to estimate the potential privacy impact. Without having access to Layer 3 and 4 meta data, the main challenge of this approach is the identification of potential requests being sent to the search engine. We use a linear regression-based approach to identify candidate packet sequences for the feature extraction. The evaluation is done in an optimal environment (reduced WiFi-traffic) to determine a first tendency and performed using three search engines. In total four different user identification/verification approaches are utilized: M1 identification using a neural network, M2 identification using Manhattan distance, M3 identification using Euclidean distance and M4 verification using a one-class support vector machine (SVM). Our results show a classification performance for 10 different test subjects is ranging from 13.2% using the one-class SVM (M4) to 64.1% using the neural network (M1) for the identical search engine. In comparison to a group of five test subjects it can be seen that M1 provides more scalability in comparison to M2, M3 and M4.*

*In addition to that, we present potential countermeasures which could help to increase the privacy of the users of a search engine.*

## Introduction and Motivation

With the multitude of mobile devices the communication via wireless networks is an essential part of our everyday life. However, due to the nature of such a shared medium, everyone in the range of the network is able to observe the communication. In order to mitigate the resulting privacy and confidentiality issues, the networks as well as the communication are usually encrypted. While a strong encryption is usually suitable to ensure the confidentiality of the communicated information, this is not necessarily the case for the privacy of individuals or observability and link-ability. The concepts of privacy-by-design and privacy-by-default increasingly gain attention. On the other hand, more and more convenience features are implemented into applications and web services. Albeit the common usage of encrypted communication channels, an analysis of the user behavior with the intention of user identification or verification is still possible as shown by Whiskerd et al. [7] adapting the behavioral biometrics approach of keystroke dynamics [6] for the search engine search suggestion functions. However, in order to reveal the potential violation of the privacy of the user, direct access to the network communication is necessary. Especially with the possibility to observe the behavior of network users potential privacy risks might arise.

In summary, the paper addresses the following objectives:
(1) The first objective of this paper is the circumvention of the limitation of the approach of Whiskerd et al. [7] by capturing and analyzing the encrypted WiFi traffic in order to be able to extract the biometric modality of keystroke dynamics [6] as well. This additional link-layer encryption blocks the ability of analyzing the layer 3 and 4 meta data contained within the TCP/IP headers of the TLS encrypted network traffic. As a result, only the timestamps of the captures, the packet size and the source and destination layer 2 addresses are visible to the observer. In addition to that, requests of a specific search session cannot be easily differentiated from other traffic originating from the same device. On the other hand, access to the broadcast traffic is significantly easier and possible even over longer distances by using antennas with higher gain.
(2) Our second objective is twofold: firstly the identification of the respective search engine interactions in order to allow for secondly deriving features suitable for a biometric identification of the users. For the latter, the objective is the assessment of the biometric matching performance as an indicator towards the resulting privacy impact. In our analysis we perform a training and testing for two tasks: the search session detection and the user identification. We employ linear regression in order to determine candidate packets for a search engine session. Afterward, up to 21 keystrokes are taken into account to create the 20-dimensional feature vector. For shorter search strings a padding is employed in order to ensure an identical dimensionality of the feature vector. The biometric identification is studied, performed and compared using four different classification approaches:

- M1: neural network,
- M2: Manhattan distance,
- M3: Euclidean distance,
- M4: One-Class Support Vector Machine (SVM).

Our experiments with 5 and 10 test subjects indicate that the identification of users in encrypted WiFi communication is indeed possible. However, in terms of the scalability only the neural network M1 seems to be slightly more robust with an identification performance of 79.07% for a test group of 5 users and 64.1% for a test group of 10 users for the same search engine. As we show the overall risk and privacy impact, a first set of countermeasures

are summarized.
The paper is structured as follows: first we summarize he most relevant state-of-the-art in keystroke dynamics as well as the analysis of this biometric modality within encrypted network traffic. Afterwards, we introduce our concept and approach for detecting users by observing WPA2 encrypted WiFi traffic, followed by a description of our experimental setup. In the second to last section the results are being shown and discussed. Subsequently we summarize our findings and a set of first countermeasures before outlining potential future work.

## State-of-the-Art

This section describes the general concept of keystroke dynamics as a behavioral biometrics discipline as well as the extraction of such biometric characteristics from encrypted network traffic.

### *Keystroke Dynamics*

Biometric characteristics can be used to uniquely identify individuals. These characteristics are distinguished by their uniqueness, constancy, measurability and universality. One of them describes the typing behavior on keyboards, also known as keystrokes dynamics. This means that every individual can be uniquely identified based on their characteristic typing pattern.
One of the first papers dealing with this topic is the paper of R. Stockton Gaines et al. [4]. They conduct various experiments with a small group of seven people and discover signs of individual typing patterns. They use the time between two keystrokes in order to identify a user.
The survey paper of Pisani et al. [6] summarizes a set of five features derived from key presses (key down, D) and key releases (key up, U):

- DU1 - from a key-press to the release of the same key (dwell time),
- DU2 - from a key-press to the release of the next key,
- UD - from a key-release to the next key-press (flight time),
- DD - from a key-press to the next key-press,
- UU - from a key-release to the next key-release.

Based on those basic features several machine learning techniques can be utilized in order to create templates for users and for performing verification or identification tasks. However, in network traffic only a subset of those features can be observed as summarized in the following subsection.

### *Keystroke Dynamics in Network Traffic*

Our paper is based on the work of Whiskerd et al. [7] analyzing the impact of search suggestion functions of modern search engines on a users privacy based on observable keystroke dynamics. In their attempts, they use the two browsers Google Chrome and Mozilla Firefox, as well as the search engines Google, Qwant, DuckDuckGo and Ecosia. They also conduct tests on different keyboards, a hardware keyboard and a software keyboard (Android system). These two keyboards differ in the way that a character input is detected by the system and thus, the time when it is visible within the network communication caused by the search suggestion function. In the case of hardware keyboards, characters are entered on key-down events (pressing the key down). With software keyboards, on the other hand, character input occurs on key-up events (releasing the key). With respect to the work of P. H. Pisani et al. [6] this resembles DD and UU times.
To investigate the tendencies for WiFi traffic, in our work we focus only on hardware keyboards, minimize the network traffic of other applications and capture the time between two key-down events (DD). Each additional character results in a new search request which is visible on the network layer. Depending on the level of encryption either the full search string or just meta-data is visible to an observer on the network. Overall the results of Whiskerd et al. [7] show that an identification of users is indeed possible by observing network traffic caused by search suggestion functions.
With the help of WLAN sniffers, such as airodump [11] and wireshark [12], we can analyze data traffic in encrypted wireless networks. These programs allow us to detect wireless computer networks and to intercept the transmitted data without any knowledge of the pre-shared secrets for accessing the communication contents.
To the best of our knowledge, there are only a few other works, e.g. [5], addressing the research challenge of user identification based on keystroke dynamics in encrypted networks.

Additionally, the work in [7] was extended in the Bachelor's thesis of Nicklas Krtge [8] supervised by Jana Dittmann and Christian Kraetzer, focusing on an estimation of the amount of behavior information that can be derived at different locations (on the device, its WiFi access point, the nearest gateway/router, an gateway outside the users LAN and the search engine provider) when analyzing the keystroke information transmitted by the Search Suggestion Function (SSF) of one prominent internet search engine. A short English summary of the contents of this Thesis is provided at [9].

## Concept for Identifying Users by Observing Encrypted WiFi Traffic

Our concept exploits the intrinsic property of WiFi networks of being a shared communication medium as depicted in Figure 1. While Alice and Charlie are using the search session Bob, Eve is able to observe all sent and received WiFi frames as long as she is in the range of the wireless network. In case of encrypted networks, she is able to see the layer 2 addresses of the communication partners within the WiFi network (Alice, Charlie and the

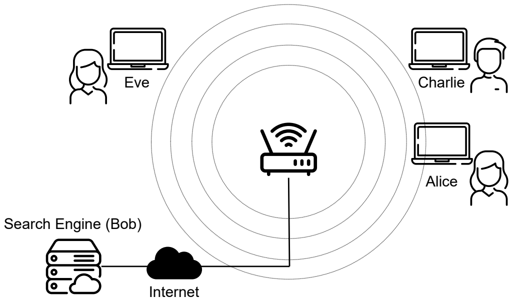


*Figure 1. General Concept for Identifying Search Engine Users in Encrypted WiFi Traffic*

access point), the length of the frame as well as some other metadata for the wireless network. In contrast to [7] all layer 3 and 4 meta data is encrypted and cannot be evaluated in order to detect the session.
Thus, the overall concept consists of the following steps:

1. Template Creation
2. Search Query Packet Sequence Detection (Search Session Segmentation)
3. Feature Extraction (with Padding)
4. Classification (Identification)

To research the optimal case and the tendency, the template creation is covered separately because it is not performed on the WiFi-Data. Instead, a custom-built JavaScript-based tool[1] is utilized in order to record keystrokes accompanied with a ground truth. In real-world use-cases the training must be performed by observing the network as well - which likely results in less accurate templates and thus, higher error rates, which should be researched further in future work.
Furthermore, for the development of the approach, at first the WiFi traffic is analyzed by additionally capturing the network communication on a client computer executing the search queries. The respective capture files are correlated based on synchronized times of the computers real time clocks for labeling purposes. Based on this information a search engine specific window for packet sizes and the resulting overhead of 50 bytes for the encrypted WiFi network is determined. In particular frame sizes between 230 and 310 bytes are empirically determined for the three search engines. The remainder of this section describes the steps two to four of our concept describing the selection of potential search engine sessions, the feature extraction and the selected classification approaches.

### Search Query Packet Sequence Selection

We use linear regression to detect the potential search engine session. At first the candidate packets are selected if they are in a length range between 180 and 260 bytes (not counting the overhead of 50 bytes), all other packets are ignored. Those values have been determined empirically. In addition to that, additional assumptions are made in order to filter the network traffic: two keystrokes are at least 50 milliseconds apart and the packet size between two packets of a session is increased by 0 to 3 bytes. An “increase” of 0 bytes is considered due to the utilized TLS session between Alice/Charlie and Bob, which might result in the same packet length even with an additional character in the search string. A session is terminated if no additional keystroke has been detected within two seconds or if the observed packet length is increased by 50 bytes. Sessions with less than five detected search packets are discarded.

### Feature Extraction (with Padding)

Based on each session, a 20-dimensional feature vector is extracted. Each feature vector represents the time offsets of 21 detected search requests within the session. If less than 21 keystrokes are recorded, a padding of the missing values is performed. We implement the padding by determining the average value of the last three recorded non-zero values of the feature vector. This process is repeated until the designed length of the feature vector is reached.

### Classification Approach

Within the scope of the classification phase of this paper we use three identification methods and one verification approach. For identification, we use a neural network as well as two common distance measures, namely Manhattan and Euclidean distances. The verification approach is implemented by using one-class support vector machines (SVM) for each user. This is considered as a verification approach as the SVM trained for one user must be selected. The resulting score indicates whether the captured data matches the data used for training the SVM. Of course, an identification is possible using this approach as well by systematically evaluating all trained SVMs. The following subsections describe each utilized classification approach in more detail.

#### *Neural Network (M1)*

The neural network based classifier is created on the foundation of Keras[2] and TensorFlow[3]. The network consists of three layers as depicted in Figure 2. At the input layer the 20 dimensional feature vector as well as a label (one-hot encoding) is fed into the neural network. The first layer of the network uses those 21 values as an input to set the output of 20 neurons using the Rectified Linear Unit activation function ’ReLU’ of the Keras framework. The second layer has 10 output layers and is using the same activation function as the first layer. The third layer, which is also our output layer has 10 neurons for the extended test set and 5 neurons for the initial test set. Each neuron represents the assigned identity. In contrast to the first two layers, the activation function ’softmax’ of the Keras framework is utilized.
The main disadvantage of the neural network based approach is that the network needs to be retrained and potentially reconfigured on the addition of another user.

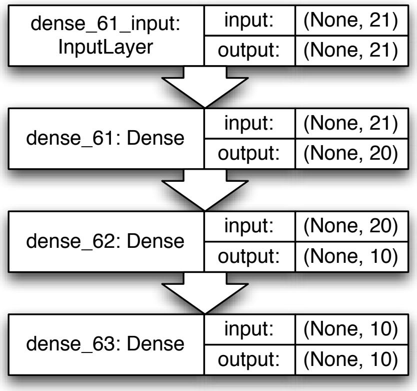


***Figure 2.*** *Structure of our Neural Network for User Identification (M1)*

#### *Manhattan Distance (M2)*

Distance measures are a more traditional approach in biometric systems. The main advantage is that the decision making

[1] Created by the student team as mentioned in the acknowledgements section

[2] https://keras.io/

[3] https://www.tensorflow.org/

process is rather easy to understand and that particular thresholds could be used set in dependence of the design goal of the biometric system (convenience vs. security). A very simple distance measure is the Manhattan distance (also known as city block distance).

For our experiments we utilize the scaled Manhattan distance which is described by Arajo et al. [1]. The classifier (in this case the template) is created by determining the mean time vector for the training data $y_i$ for the i-th keystroke in the template. During the identification the observed keystroke $x_i$ is compared with the corresponding value of the template $y_i$. The score $s$ is determined using $s = \sum_{i=1}^{20} \frac{|x_i - y_i|}{a_i}$ with $a_i$ being the scaling factor for the specific keystroke. Finally, the particular template with the lowest score $s$ is selected as the assigned identity.

### *Euclidean Distance (M3)*

The Euclidean distance is similar to the Manhattan distance in the previous subsection. At first, a template is created within the 20-dimensional feature space based on the collected training data. For each identity, the center of the data points within the feature space is calculated as the corresponding template.

During the classification the distance score $s$ is determined as follows: $s = \sqrt{\sum_{i=1}^{20} (x_i - y_i)^2}$ with $y_i$ being the i-th keystroke of the template and $x_i$ the i-th observed keystroke of the sample. Identical to the Manhattan distance, the particular template with the lowest score $s$ is selected as the assigned identity.

### *One-Class Support Vector Machine (M4)*

For the SVM-based verification we select the approach described by Yu and Cho [2] with a radial basis function as the kernel.

In the training phase a one-class SVM is created for each test subject. For the classification we use the ’decision_function’-method which returns the signed distance to the separating hyperplane of the SVM. Positive values represent inliers whereas negative valuers represent outliers of the trained model. Similar to the distance measures we use this particular value as a raw matching score. However, in this case larger values indicate a better match of a specific template (SVM). Thus, we select the highest score, i.e. the largest positive distance to the decision boundary represented by the separating hyperplane, to select the identity for a presented sample.

## Experimental Setup

This section describes the measurement setup as well as the gathering of training data for our experiments. It is designed as an optimal case to study the overall tendency.

### *Measurement Setup*

The measurement setup deviates from the general concept depicted in Figure 1. In fact the setup shown in Figure 3 utilizes one client computer system equipped with two WiFi adapters running Ubuntu Linux 20.04 LTS in order to capture test data for 7 of 10 (P3-P7) test subjects. The first, integrated, WiFi adapter is utilized by Alice for communicating with the search engine Bob on the internet via the encrypted WiFi network. On the second, external network adapter (TL-WN823N) the WPA2 encrypted WiFi traffic is recorded by Eve using airodump from the Aircrack-NG[4] package utilizing the monitor mode. In Addition to that, ground truth data is collected by Alice as well using Wireshark[5] in conjunction with exported SSL (pre-)master secrets from the web browser. The overall intention of this setup is the synchronization of the time source which greatly improves our ability of debugging the approach.

Additionally, test data for three users (P1-P3) running Microsoft Windows 10 (home version and default settings) is recorded using two different machines. The reason for the deviation in the test protocol was the limited hardware availability during the experiments.

All experiments are performed using Mozilla Firefox 76.0.1 (default settings) as the web browser for Alice with no additional applications running. In total three different search engines are evaluated:

- S1: https://www.google.com/
- S2: https://duckduckgo.com/
- S3: https://www.qwant.com/

For each search engine each test subject (P1, ..., P10) is asked to perform 10 search queries. The first 5 queries (Q1-Q5) are performed using the identical search string “Schreibtisch”, the remaining 5 queries are performed with search strings of a variable length as summarized in Table 1. All test participants speak German, use the identical set of search strings and are used to the German keyboard layout. As all search strings with the exception of Q9 are shorter than 21 characters, the introduced padding ap-

**Table 1: Overview of training samples and Utilized Search Strings for testing Query used for all user P1-P10.**

| Query | Search String |
|---|---|
| Q1-Q5 | Schreibtisch |
| Q6 | Wetter |
| Q7 | Whats(app) |
| Q8 | W.H.A.T. |
| Q9 | Eclipse IDE for Java Developers  2020-03 |
| Q10 | Was machst du?! |

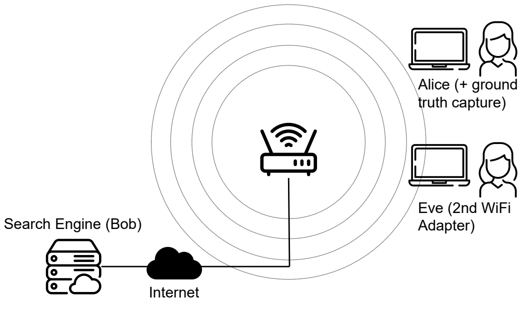


*Figure 3. Measurement Setup for our Experiments*

[4] http://aircrack-ng.org/

[5] https://www.wireshark.org/

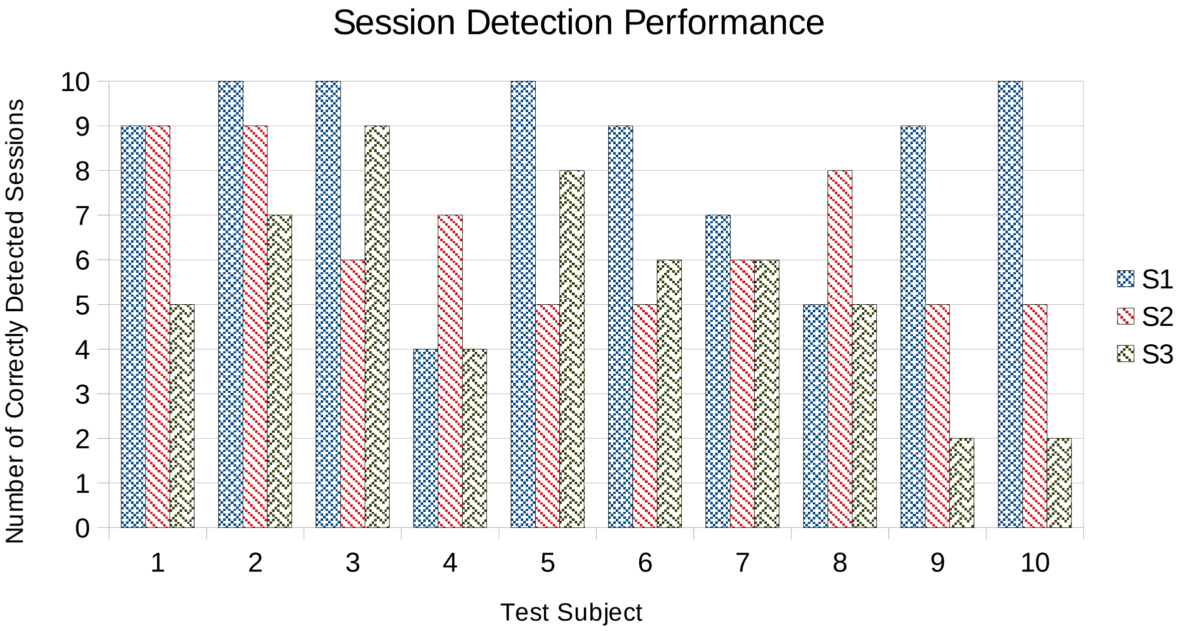


*Figure 4. Search Session Detection Performance for each of the 10 Test Subjects and Utilized Search Engine (S1 Google, S2 DuckDuckGo, S3 Qwant)*

proach is utilized in almost all test cases.
The first five test subjects (P1, ..., P5) represent the initial test group. The test results for this group are shown and discussed in the results section as well in order to visualize the scalability of the suggested identification approach.

### Training Data and Classifier Training

For user identification the training of the classifiers ground truth samples are gathered for each test subject P1-P10 using a purpose-built JavaScript code which is completely independent of potentially utilized search engines. The code is designed to display random combinations of search terms which a user should type independently to the test data queries (see Table 1). While typing, the times between observed key presses (down-down-times) are recorded. This process is repeated 50 times resulting in 50 different typing sequences in order to determine the intra-person-variance of the typing behavior of the test subject.
For M2, M3 and M4 the full set of 50 training samples is used to train the classifiers. For the training of the neural network M1, we perform a percentage split using 70% of the recorded sessions for training, whereas the remaining 30% for the validation. Overall the neural network is trained over 500 epochs in conjunction with an early stopping as soon as an over-fitting is detected.

## Results and Discussion

The full test results rely on search queries performed by the ten test subjects. In addition to that, a subset of five test subjects is used for the initial experiments. Both results are compared with each other in order to estimate the scalability of the proposed approach.
First the results for the search session detection can be summarized as follows: Towards the search session detection, an attempt is considered successful if at least 80% of the packets are correctly detected and no more than 20% of false positives are included in the session. The session detection yields for all users P1-P10 results between 83% for S1 (Google), 65% for S2 (DuckDuckGo) and 54% for S3 (Qwant) as shown in Figure 4 for each individual test subject. The per-user results differ significantly. For four test subjects all S1 sessions are correctly detected. In contrast to that, for two test subjects only 20% of the S3 sessions are detected. Thus, we can assume that the approach shows the first tendency and is in general suitable, but requires additional tuning towards the specific search engine characteristics.
The search session detection performance is independent of the number of known users as it is purely performed on the observed packets. After this step, the features are extracted and the biometric evaluation is performed.
The overall identification performance for the subset of five test subjects is summarized in Table 2. The results show that the best identification performance is achieved using the neural network, followed by the more traditional biometric approach utilizing the Euclidean distance measure. The Manhattan distance and the one-class SVM perform significantly worse. However, throughout the

**Table 2: Overall Identification Performance (P1-P5), best results per classifier highlighted in bold face**

| Classifier | S1 | S2 | S3 |
|---|---|---|---|
| M1 - Neural Network | **79.07** | 66.67 | 69.70 |
| M2 - Manhattan | **39.53** | 38.89 | 36.36 |
| M3 - Euclidean | 60.47 | **61.11** | 57.58 |
| M4 - SVM | 32.56 | 36.11 | **42.42** |

three different search engines, the two distance measures yield the more consistent performances. Whereas for the neural network especially for S1 a significantly better performance is achieved in comparison to S2 and S3. For the SVM the complete opposite can be observed with the worst performance being achieved in conjunction with S1.
For the full set of then test subjects the results are summarized in Table 3. Here, the performance is significantly lower. In comparison to the test set of five test subjects, M1 has lost roughly 15 percent points in identification performance for S1. The decrease is in a similar magnitude for S2 and S3. Nevertheless, the results for M1 indicate a better scalability in comparison to M2, M3 and M4 which have lost more than 50% of their original detection accuracy in comparison to the smaller test group. In addition to that, the results for M2 are less consistent throughout the search engines.

**Table 3: Overall Identification Performance (P1-P10), best results per classifier highlighted in bold face**

| Classifier | S1 | S2 | S3 |
|---|---|---|---|
| M1 - Neural Network | **64.1** | 53.5 | 55.7 |
| M2 - Manhattan | 13.2 | 12.6 | **13.6** |
| M3 - Euclidean | 22.2 | 23.8 | **27.1** |
| M4 - SVM | 13.2 | 12.7 | **17** |

## Potential Countermeasures for Strengthening Privacy

The root-cause for the possibility of performing keystroke dynamics even in encrypted traffic is the behavior of the search engine. Using common web technologies usually in default settings, a request is sent to the search engine after every single keystroke. This allows for determining a session based on the gradually increasing length of the request packet as described in the Section on the search query packet sequence selection.
A first set of potential countermeasures are summarized in Table 4, derived from ideas known in anonymity mechanisms such as techniques in mixes or Onion routing (see e.g. [10]) which have a well known implementation in the TOR network. So-called dummy traffic in such networks would of course also be a possible general countermeasure against profiling provided by the used network infrastructure itself, but would cause a significant overhead of network traffic, resulting in much higher connection costs. For this reason it is not included in our first set of recommended countermeasures.

The main objective of the countermeasures is to retain the functionality of the suggestion function while breaking the possibility of deriving keystroke biometrics. The proposed countermeasures can take place on either user's computer, as an add-on PET (privacy-enhancing technology) application, or on the side of the search engine provider, essentially following privacy-by-design principles from early design stages. The latter could be advantageous for the search engine provider since it complies with the GDPR [3] article 9, avoiding to request the user's explicit consent for the acquisition of biometric data for identification or profiling. Thus, implementing privacy protection techniques ensure Article 25 (data protection-by-default) which states: "the controller (search engine provider) must implement appropriate technical mechanisms and approved certification mechanisms to demonstrate compliance with the requirements in some specific GDPR articles" [3]. Moreover, the proposed design features in Table 4 can be used for advertising purposes as a proven privacy-friendly service.

**Table 4: Potential Server- and Client-Side Countermeasures**

| Countermeasure | Server-Side (Bob) | Client-Side (Alice) |
|---|---|---|
| No Search String Suggestions (deteriorated user experience) | Disable service | Disable JavaScript for search engine |
| Pooling | Pooling in JavaScript Code | Limit Requests per Time-Interval/Pooling of Requests |
| Random Delay | - | Delay of Requests |
| Normalization | - | Fixed Request Time Window, e.g. Once every 2s |
| Padding | Random Length Padding Element | Enforce Fixed Request Length |

## Conclusions and Future Work

In this paper, we shed light on privacy implications of search engine suggestions in encrypted WiFi networks. The results show that a limited number of users can be differentiated based on inter-packet times during the typing process of a search query. The concept shows that a limited biometric recognition derived from keystroke dynamics is possible in an encrypted data stream with almost no accessible metadata. The comparison of test groups with five and ten test subjects shows a limited scalability of the approach. However, at least the neural network is still performing acceptable for the doubled number of test subjects achieving up to 64.1 percent of correct identifications in conjunction with the Google search engine. The first party providers of course are in a much better situation of capturing the typing behavior as they do not need to detect the search engine session itself. For a third party attack such as Eve, our first approach of course is still a limited setting as the ground truth for the template generation is captured with a local JavaScript-based approach. This is a realistic assumption only for the search engine provider as first party capture but is limited for third parties. The presented results show a first tendency and can be used as starting point for a broader assessment.
As our suggested approach does not rely on identifiers, such as MAC addresses, it allows for circumventing common mechanisms like address randomization and cross-device tracking as long as the input method remains unaltered, e.g. a physical keyboard for all utilized devices.
In future work the impact of the suggested countermeasures should be evaluated and further known approaches from anonymity and PET (Privacy Enhancing Technology) should be

elaborated. Furthermore, the number of test subjects should be increased in order to investigate the scalability of the approach in more detail. The most important task in future work is the improvement of the search session detection as any detection errors will eventually impact the identification performance. This is also necessary in order to determine the suitability of the suggested countermeasures.

In general our work also should motivate to analyze biometric typing behavior in other encrypted protocols. Here we want to point out that collaboration platforms and tools, e.g. for joint collaborative writing, also have the potential to leak such information by sending individual keystrokes. This creates observable traffic between users that could be analysed at all clients sides (raising also multi-party privacy issues) as well as in the network traffic. If no countermeasures are applied this also would allow for a general capturing of keystroke dynamics as presented in our work on the example of search engines.

## Acknowledgments

The privacy related issues are motivated and partially supported by "AMBER - enhAnced Mobile BiomEtRics" Project ID: 675087, European Unions Horizon 2020 research and innovation programme within PhD research work. Further the work about the protocol analysis has been funded in part by the German Federal Ministry for Economic Affairs and Energy (BMWi, Stealth-Szenarien, Grant No. 1501589A).

The paper presents joint work of all co-authors, in particular with following individual shares: The paper is an extension of a student project in the summer term 2020 of Aamir Shakir, Alexander Ziemke, Mohamed Abdelrazek, Hannes Stuetzer and Dominik Blut based on a task designed and assigned by Mario Hildebrandt and Jana Dittmann extending the idea of the experiments in [7], as described in the students' task description. Furthermore, both supervised the students work and gave feedback. Jana Dittmann originally proposed in [7] the idea and and was as main authors responsible as contact to EI and for structuring, collecting, coordinating and integrating all inputs from the co-authors. Jana Dittmann, Mario Hildebrandt, Christian Kraetzer, Kevin Lamshoeft contributed in the general idea of the extension of [7] to wireless protocols by using e.g. WiFi or wireless keyboards and contributed in the discussion and setup of the student task design.

Figures 1 and 3 have been re-sketched by Aamir Shakir based on the initial draft of Mario Hildebrandt. The implementation and experimental study of the search query packet sequence selection (search session detection) and the user identification (classification approaches) is the result of the student project with initially 5 users. The extended evaluation with the final set of 10 users has been performed by Aamir Shakir. As the developer of the algorithm for automatically identifying search packets based on their packet length, Alexander Ziemke was again available as an expert in this paper to answer queries about applying the algorithm to the new datasets, monitoring and validating it, and optimizing its implementation. In Addition to that, Dominik Blut was at hand to solve potential problems regarding the recognition of search engine packet patterns that might have been changed since the initial creation of the so called ground truth in the initial work of the students. The potential countermeasures are discussed and enhanced with all authors and summarized as joint work of Mario Hildebrandt, Salatiel Ezennaya-Gomez and Kevin Lamshoeft in the Countermeasures section. Overall inputs end enhancements regarding the paper structure and final improvements of the paper have been provided by Jana Dittmann and Christian Kraetzer. Work in the Conclusions and Future Work section in respect to collaborative working platforms and multi-party privacy issues was motivated from Jana Dittmann from her first tests in joined work with Robert Altschaffel and Christian Kraetzer.

## References

[1] Lvia C. F. Arajo, Luiz H. R. Sucupira, Miguel G. Lizrraga, Lee L. Ling, Joao B. T. Yabu-uti. User Authentication through Typing Biometrics Features, Proc. International Conference on Biometric Authentication, pg. 694-700 (2004).

[2] Enzhe Yu, Sungzoon Cho. GA-SVM wrapper approach for feature subset selection in keystroke dynamics identity verification, Proc. International Joint Conference on Neural Networks, pg. 2253-2257 (2003).

[3] European Parliament, Regulation (EU) 2016/679 (General Data Protection Regulation), (2016).

[4] R. Stockton Gaines, William Lisowski, S. James Press, Norman Shapiro. Authentication by keystroke timing: some preliminary results, Rand Rep. R-2526-NSF, (1980). ISBN 0-8330-0246-5

[5] Robert Koch, Gabi Dreo Rodosek. User identification in encrypted network communications, 2010 International Conference on Network and Service Management, pg. 246-249 (2010).

[6] Paulo Henrique Pisani, Ana Carolina Lorena. A systematic review on keystroke dynamics, J Braz Comput Soc, 19, 573-587 (2013).

[7] Nicholas Whiskerd, Nicklas Körtge, Kris Jürgens, Kevin Lamshöft, Salatiel Ezennaya-Gomez, Claus Vielhauer, Jana Dittmann, Mario Hildebrandt. Keystroke biometrics in the encrypted domain: a first study on search suggestion functions of web search engines, EURASIP J. on Info. Security 2020, 2, https://doi.org/10.1186/s13635-020-0100-8, (2020).

[8] Nicklas Körtge. Potenzielle Verletzungen der Privatsphäre durch unzweckmäßige Auswertung von Keystroke Dynamics in Netzwerkdatenströmen. Bachelor Thesis, Dept. of Computer Science, Otto-von-Guericke University Magdeburg, Germany, 2020.

[9] Nicklas Körtge. How to extract Keystroke Biometrics from encrypted Google search traffic. Available online at: https://nicklaskoertge.medium.com/how-to-extract-keystroke-biometrics-from-encrypted-google-search-traffic-2772aaa47ed9, website request February 12th, 2021.

[10] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM. 24 (2): 8490, 1981.

[11] airodump-ng [Aircrack-ng]. `https://www.aircrack-ng.org/doku.php?id=airodump-ng`. Accessed 22-02-2021.

[12] Wireshark Go Deep. `https://www.wireshark.org/`. Accessed 22-02-2021.

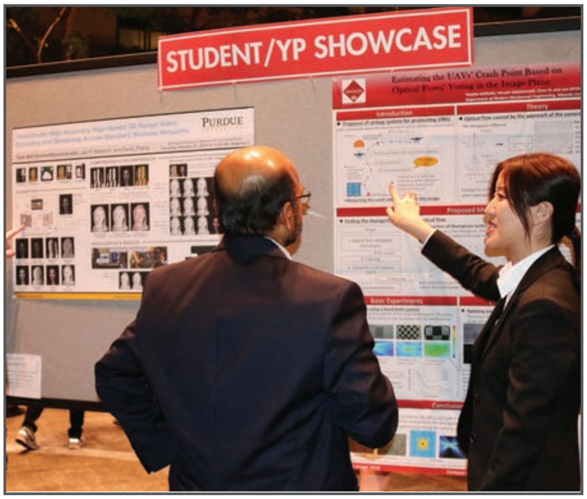
## JOIN US AT THE NEXT EI!

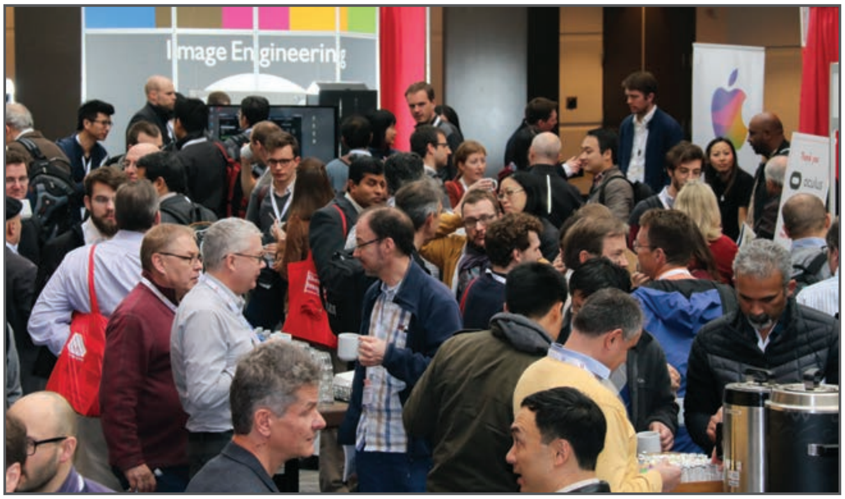
IS&T International Symposium on

# Electronic Imaging

SCIENCE AND TECHNOLOGY

*Imaging across applications . . . Where industry and academia meet!*





- AR is a fundamental and transformative interface and platform.
- 20+ years from my initial working demo to today's AR displays
- Optimistic that it won't take another 20 to reach ubiquitous consumer AR!

• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

**www.electronicimaging.org**

IS&T
imaging.org