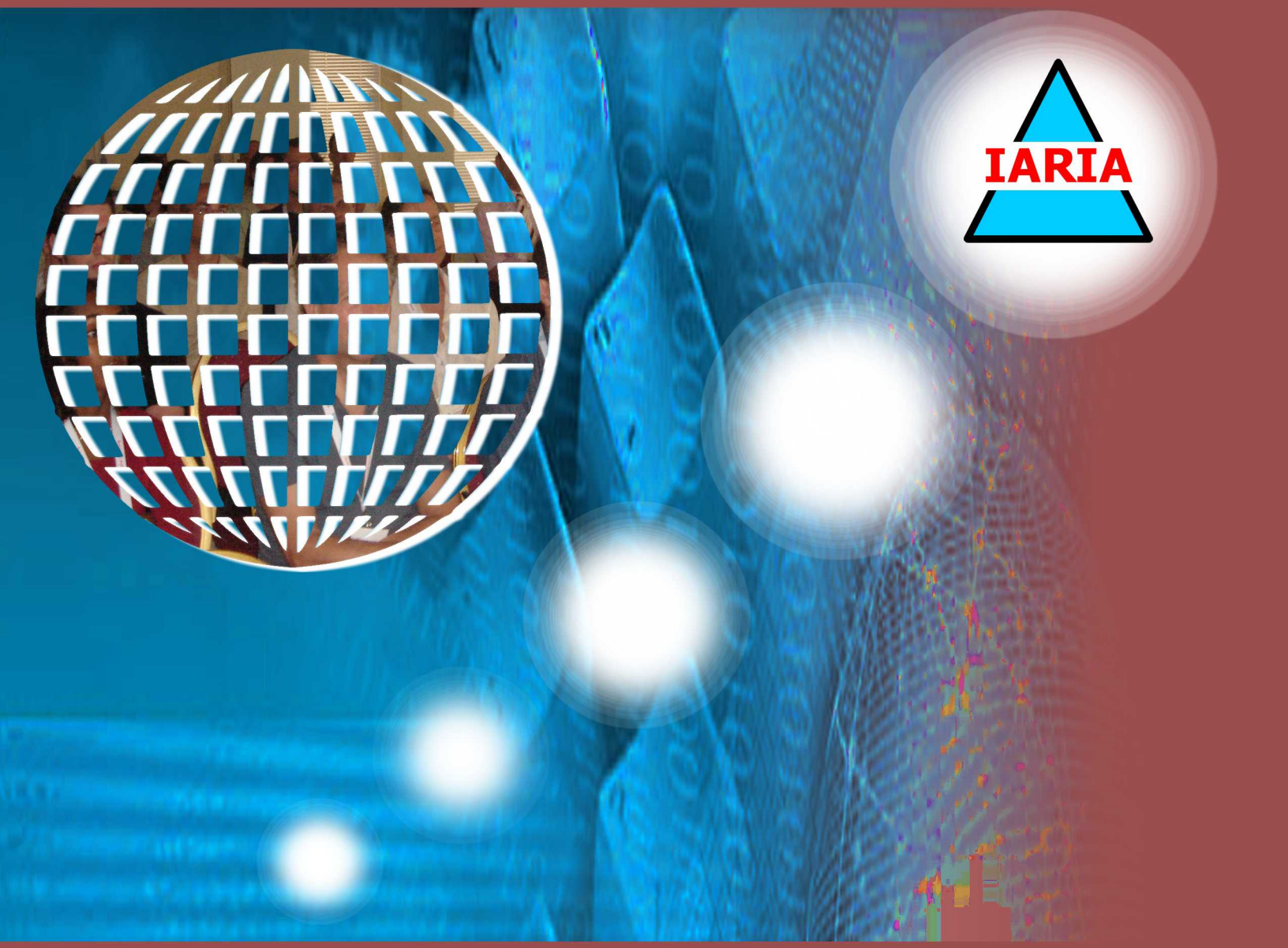
International Journal on Advances in Security



2024 vol. 17 nr. 3&4

The *International Journal on Advances in Security* is published by IARIA.
ISSN: 1942-2636
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Security
Volume 17, Number 3 & 4, 2024

Pedro Sousa, University of Minho, Braga, Portugal
Miroslav Velev, Aries Design Automation, USA
Cong-Cong Xing, Nicholls State University, USA

International Journal on Advances in Security
Volume 17, Numbers 3 & 4, 2024

CONTENTS

AI-Driven Analysis for Network Attacks: Enhancing IDS Alerts and ACL Integration

Nader Shahata
Center for Strategic Cyber Resilience Research and Development
National Institute of Informatics
Tokyo, Japan
e-mail: nader@nii.ac.jp

Hirokazu Hasegawa
Center for Strategic Cyber Resilience Research and Development
National Institute of Informatics
Tokyo, Japan
e-mail: hasegawa@nii.ac.jp

Masahiko Kato
Department of health data science
Juntendo University
Tokyo, Japan
e-mail: m.kato.ug@juntendo.ac.jp

Hiroki Takakura
Center for Strategic Cyber Resilience Research and Development
National Institute of Informatics
Tokyo, Japan
e-mail: takakura@nii.ac.jp

***Abstract*—Due to the widespread deployment of digital systems, and the increasing complexity of cyber threats, it has become crucial to us to secure our resources in computer connected systems. Access Control Lists (ACLs) are fundamental frameworks that govern the authorization and authentication processes that occur in our network. Essentially, ACLs are a set of rules that define users who have permissions to access particular resources. Furthermore, ACLs indicate whether a user's access will be permitted and what specific actions they will be able to perform. Access control lists play a vital role in the security and confidentiality of sensitive information and resources. However, the emergence of artificial intelligence has the ability to transform the process of access control lists which may result in securing our network. When the system manages the network traffic with the generated ACL, it will enable the network analysts to track certain threats first without having to monitor all network traffic. This method will allow for more efficient threat detection and analysis ending up with saving time and resources. In this paper, we will discuss the usefulness of artificial intelligence and its role in generating access control lists and the consequences of using such technology in securing our network.**

Keywords-Access Control List; Cyber Security; Network; Intrusion Detection; Artificial Intelligence.

I. INTRODUCTION

This work is a follow-up to our prior work "AI-based Approach for Access Control List Management", published in the proceedings of SECURWARE2023 [1]. Access control models are crucial components in the field of information security, ensuring that only authorized individuals or entities can gain entry to protected resources [2]. Over the years, advancements have been shifted towards access control systems. One such transformation is the integration of AI and access control models. AI, with its ability to mimic human intelligence and to make informed decisions based on a massive amount of data, can transform the way access control is managed, which will lead to securing our networks in return. When implemented, AI-powered access control lists can provide numerous benefits over the traditional current systems, which include dynamic access management, behavioral analysis, and adaptive learning. These models can have the ability to strengthen machine learning algorithms [8] to analyze and understand network traffic patterns, behaviors and contextual information to make real-time based access decisions. The move from manual ACLs rules to dynamic ones can lead to more accurate and adaptive access control managements. For instance, manual ACLs require human assistance in terms of rules, which are vulnerable to errors and mistakes.

On the other hand, shifting to AI techniques will help in strengthening the security measures and reducing the risks of potential unauthorized access. By learning from historical data, AIs can detect irregular patterns that identify unusual behavior that network analyst would miss. By analyzing historical data and learning from previous patterns, AI models can establish a baseline of normal behavior for users and systems [8]. Any change in the deviation from this baseline can trigger alerts and generate preventive needed actions, helping in mitigating risks and preventing security breaches; ensuring our systems to be safe by minimizing the potential risks of network attacks. This defensive approach to access control is very crucial in today's ever-evolving threat landscape, where traditional manual rule-based systems often fail in detecting sophisticated attacks; ensuring that future arising threats are recognized and dealt with correctly.

Furthermore, AI can significantly improve user experience in access control systems. The reason is with traditional models, users often face cumbersome processes, such as repeatedly entering passwords or providing multiple credentials for different systems.

The purpose of our paper is to propose an architecture that can help increase the organization’s network security by

applying AI to generate countermeasures based on ACL rules. To do so, we will discuss how feasible is AI in generating ACLs when dealing with IDS alerts. We will examine the role that IDS plays in determining potential threats, and how AI can use those alerts provided by IDS to make dynamic and corresponding ACL related rules. We will provide a feasibility on how effective the concept of AI-based ACL systems on enhancing the efficacy of network security operations through automated, context-aware access control mechanisms by the end of this paper.

The remaining of this paper is organized as follows: Section II presents the background, which discusses the current problems that this paper is aiming to solve. In Section III, we presented the Related Works where it shows the previous researches that were conducted on the field. we presented our vision on solving the drawbacks that were discussed in the background Section through an overflow figure in Section IV. The integration and merging of AI with Anomaly detection and ACL will be presented in Section V followed by our architecture proposal in Section VI along with a detailed description of its components. The feasibility of AI in managing ACLs will be shown in Section VII. Section VIII will discuss the assumptions, whereas the challenges and considerations are explained in Section IX. In Section X discusses the importance of AI in generating ACLs. The discussion part in Section XI describes how effective our proposed system can be if it is applied when detecting anomalies and generating ACLs. We end our paper with a conclusion and future work in Section XII.

II. BACKGROUND

By controlling user access and privileges, access control models can have a significant part in guaranteeing the security and integrity of digital systems. There is considerable interest in examining the potential enhancement of access control systems in light of the significant advancements in AI. This background section seeks to give an overview of AI's use in the access control paradigm, as well as its advantages, challenges, and potential future applications. The goal is to obtain understanding of the evolving status of AI-powered technology and its influence on cybersecurity by studying existing literature and industry practices [9].

The existing ACL mechanism has a number of disadvantages that are frequently encountered [14]. For instance, managing an ACL system can be very challenging. The more users, resources, and permissions there are, the harder it is to accurately manage and update ACLs. When the number of users and available resources considerably rises, ACL systems can experience scalability problems. The network administrator in this case will need to maintain a high number of access control entries, which could affect the performance of the network [14]. ACL maintenance calls for constant work and modification. The ACL needs to be manually updated if the environment changes, such as when a new user joins a workplace or when resources are added or deleted. This maintenance work can get tedious, especially in complex systems.

In traditional ACL-based systems, ACLs are inefficient because they only support explicitly declared access controls. For example, if a user has access or permissions that are unique because they belong to both the IT department and the management department, that level of access should be explicitly stated rather than inferred on belonging to both. The requirement to explicitly declare these access controls also has an impact on scalability. As the number of users, groups, and resources increases, so does the length of the ACL and the time it takes to determine how much access is granted to a particular user. Also, ACLs lack visibility because user permissions and access levels can be scattered across many independent lists. Auditing, modifying, or revoking access require testing every ACL in the organization's environment to apply the new permissions [15]. Therefore, we need a system that can deal with the previously mentioned current problems as the cyber-attacks are on the rise of being more sophisticated. The promising machine learning algorithms that are used by AI-based ACL can create wise access control decisions. It can help in dynamically determining access privileges, which involves examining a number of variables such as users' behaviors, and previous historical data [16]. This strategy can improve security by spotting and identifying anomalies.

Reducing the number of generated alerts, improving the capability to handle complex IDS alerts, and reducing the time to respond are still challenging issues for a network analyst working on an Intrusion Detection System (IDS). The reason is most modern IDS systems can generate a large number of alerts, especially in large and complex networks. The volume of these alerts can quickly overwhelm analysts, making it difficult to prioritize genuine threats that require immediate response. Our proposed system will be focusing on managing ACLs for analyzing suspicious traffic and for generating relevant countermeasures. This strategy can improve security by spotting anomalies and abnormal behaviors. Managing ACLs plays a crucial role in doing such tasks. By configuring ACLs properly, suspicious traffic can be filtered out, preventing potentially malicious packets from reaching critical network resources. Therefore, ACLs can help in identifying common attacks that have the ability to compromise the network. By analyzing ACLs on a regular basis in order to mitigate suspicious traffic, the organization's network security posture can continue to improve. ACL management is an integral part of network security because it provides the best way to prevent suspicious traffic from breaching the system. Analyzing ACLs improves the network security posture by allowing network analysts to readjust access control rules, ultimately making the network infrastructure stronger and more resistant to intruders.

Our proposed system will be relying on machine learning algorithms [7] to assist our AI-based ACL to create wise access control decisions. This strategy can improve security by spotting anomalies. AI-based ACLs will be capable of using related data to determine access decisions and generate countermeasures based on the activities of the users and possible risks that may occur when an incident may happen. By considering these generated countermeasures, the proposed system can have the ability to accurately determine the risk involved with each access request and modify access rights as necessary. The reason behind this accuracy is due to the fact that AI-based ACLs can continuously learn from access patterns and modify their decision-making models as necessary.

III. RELATED WORK

As the explosion of the digital network space has simultaneously created new forms and types of cybersecurity threats, developing sophisticated IDS systems is considered a good idea: a system that can identify attacks in real time and counteract them accordingly. Current IDS approaches rely on signature based-detection that are good at identifying known attackers but don't scale to new attack patterns. This is why machine learning and artificial intelligence have been drawn into recent studies because they are able to identify anomalous behavior and enhance IDS's responsiveness in open networks. Among the most prominent new advances in improving the IDS performance is deep learning techniques. Thus, the deep neural network-based IDS model from Zhang et al. (2019) [22] proposes several deep learning-based IDS schemes and evaluate them accordingly. These schemes include: Auto-encoder based schemes, Restricted Boltzmann machine-based schemes, Deep belief network-based schemes, Recurrent Neural network-based schemes, Deep Neural Network-based Schemes and Hybrid IDS schemes. examines several parameters and applied them into auto-encoder based IDS schemes. Their approach was relying on classifying the deep IDS Schemes based on deep learning approach associated within each. They then reviewed how each scheme will apply deep learning methods for the purpose of recognizing various intrusion types.

Similarly, Ali (2024) studied traditional IDS-based ML methods by leveraging the power of Large Language Models (LLMs) and introduces a module named HUNTGPT. Their approach proves that LLMs have the ability to play a role in the next-generation cyber security application [18].

Moreover, Zhang et al. (2019) [12] presented a research one approach called "Automated Synthesis of Access Control". Their developed a system in this paper called EASYACL uses natural language processing to provide users with the ability to create ACL rules without needing to learn complex command syntax. This represents a departure from the existed approaches, as it has the ability to interact with the system in a more intuitive and user-friendly. The extension of their work in was on conversational AI and natural language interfaces itself lies in the use of Eliza, a prototype of AI implemented natural language descriptions into ACL commands. EASYACL is an ACL-specific application and provides multi-platform outputs for devices like Cisco and Juniper.

IV. SYSTEM OVERVIEW

We propose a dynamic AI based Access Control system for solving the problems, which are explained in Section II. Our system involves the integration of AI and generating ACL for improving the network structure in dealing with suspicious traffic analysis [6]. This can lead to generate an efficient countermeasure against future similar attacks. Figure 1 shows an overview of our proposed system, it consists of five phases, which work in a sequential step-by-step order. We will describe details of each phase below.

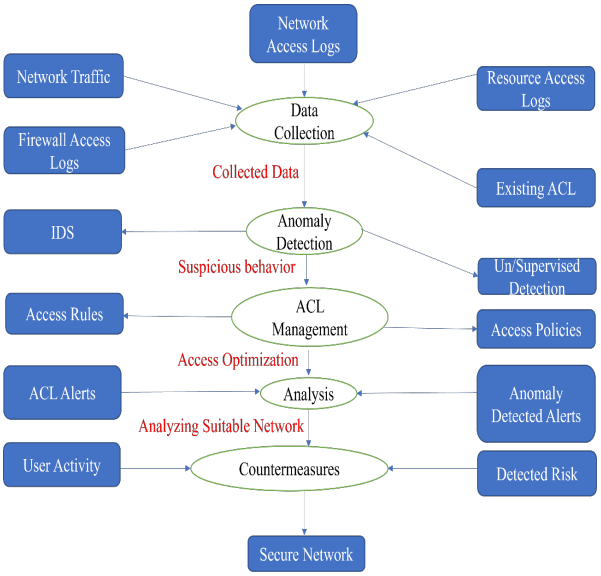


Figure 1. Proposed System Overview

A. Data Collection

This is the first phase, in which the collection of several attributes of data is required [10]. To be specific, we will be focusing on five attributes. These attributes have an edge over other candidates due to their particular concentration on certain aspects of network security. Organizations can improve their capability to identify, address, and avoid security issues by gathering and analyzing data from these sources.

These attributes include network traffic that contains all network traffic data that is observed in the organization's network, e.g., source IP-address and destination IP-address, protocol, source port, destination port. The second attribute is the firewall access logs, which are obtained and stored in

the firewall, e.g., rule numbers, protocols that have been used and the action that is taken by the firewall. The third attribute is the network access logs, (which includes the permissions of allowing or denying users from accessing the network, e.g., the user's name, connection type and connection duration). Then, we have the fourth attribute which is the resource access logs (that determine which resource are allowed or denied for specific users to access with its timestamp, e.g., accessing a financial report by a specific user in 3:00 PM). The final attribute is the applied network ACL that already existed in the system, e.g., source IP-address, destination IP-address, protocol, source port, destination port, and the action that has been taken for that rule. These attributes vary depending on the product and configuration, but are basically above formats. These attributes are all required for the next anomaly detection phase [4].

B. Anomaly Detection

In this phase, the collected data in phase one will be the input to several anomaly detection methods [5]. Currently, a lot of anomaly detection methods exist. With such existing methods, we can detect anomaly behavior from the collected data in phase one. As a typical example, we will consider IDS in detecting anomaly traffic from the network traffic data. Moreover, the applied network ACL and access logs can be used in detecting suspicious activities that are out of the authorized scope access of the network. We chose IDS in our case because it can be adapted to fit the several security configurations and the needs of organizations as well as its effectiveness when combining it with machine learning methods [8]. They can adjust to various network and system designs because of their flexibility. By inputting these data to AI, it can help in deciding whether the unauthorized activity is due to a user's fault or if it is a suspicious access attempt.

C. ACL Management

In the first and second phases, we used the existing techniques. The third phase is where AI will be applied by controlling ACL configurations to keep track of suspicious activities. Generally, this phase is the core of our architecture and is responsible for managing access rules and access policies. It will also be used to examine historical access logs and permissions data to identify patterns and their relationships. It is important to mention that the patterns and security criteria that are found here will introduce optimization algorithms or reinforcement learning approaches to enhance the ACL policy for later effective countermeasures. This will help in adjusting the ACL rules to make the network more efficient and secure.

D. Analysis

In this phase, the network analyst will evaluate the alert outcomes from the detected anomalies (in the second phase) and from the alerts that are generated from the ACL management (the third phase) to obtain a comprehensive understanding of the system security posture [6]. This posture analysis will be the input for the final countermeasures phase.

E. Countermeasures

After the network analyst evaluation, the countermeasure phase with the help of AI will prioritize the alerts and examine the likelihood and potential consequences based on the analysis result. AI will recommend the suitable countermeasures by adjusting ACLs accordingly based on those outcomes. This will help in providing more focus on the targeted resources by adjusting those resources towards the most critical security issues, instead of considering each potential threat as equally important.

V. The AI Merging of Anomaly detection and Generating Access Control Lists

AI helps in access control list (ACL) merging with anomaly identification. ACLs are used to restrict access to resources and systems based on predefined rules, whereas anomaly detection focuses on spotting patterns or behaviors that dramatically depart from the norm [3]. By employing machine learning algorithms [8] to analyze massive volumes of data and spot strange patterns and behaviors, AI can enhance anomaly detection [7].

AI also can play a major role in real-time monitoring IDS alerts. When anomalies (such as unusual traffic patterns, malware or exploited traffic caused be network attacks) are detected, AI has the ability to analyze and examine such patterns and creates specific Access Control Lists (ACLs) to prevent or restrict access according to these incidents.

ACLs, on the other hand, will get evolved and updated very conveniently by the help of AI by learning the network behavior and historic past attack patterns. This will also help in keeping the network security measures in sync against zero-day attacks as well [27].

An AI model may learn what is considered typical behavior and recognize variations that may reveal potential security issues or anomalies by being trained on previous data samples. Identifying unauthorized access attempts and odd system activities will be easier for the network analyst for examining the network's security position. AI can assist in automating the management and enforcement of access restrictions in the context of access control lists. AI algorithms are able to decide what permissions are appropriate for certain users or groups of users by examining user behavior and previous access patterns [6]. This will also simplify the management of ACLs [12], particularly in complicated systems with lots of users and resources. Access control lists and anomaly detection can be used to offer a more complete security solution. AI system's detection of anomalous behavior may result in updates to access control lists (ACLs) to restrict access or notifications for further enquiry. By dynamically modifying permissions based on in-the-moment abnormalities, this integration makes it possible to take a preventative approach to

security, lowering the likelihood of unauthorized access and malicious activities.

Overall, AI can enhance security posture, automate procedures, and increase the effectiveness of permission management in complicated systems by combining anomaly detection and access control lists.

VI. PROPOSED ARCHITECTURE

Before presenting our proposed system in this section, it is important to mention the idea of iteration. Our system is based on the alerts and the generated ACL rules that will be the fundamental concept behind our architecture to work properly. Moreover, it will validate the accuracy and effectiveness when they will be examined by a network analyst. This iterative process helps refine the architecture's performance and will enhance the overall system's output. The proposal of our architecture is as follows.

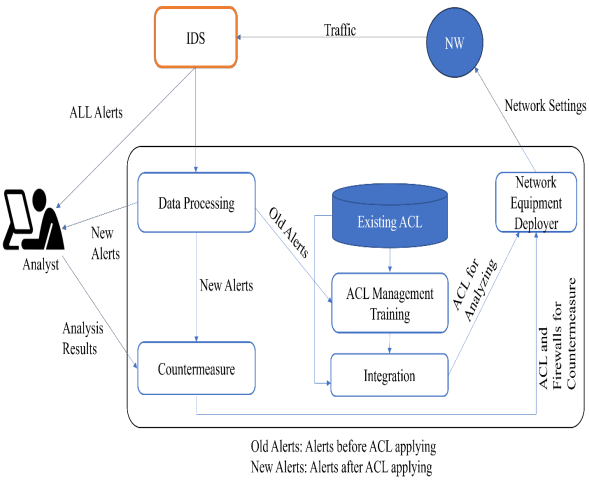


Figure 2. Proposed System

When matching the discussed overflow in Figure 1 with the system proposed in Figure 2, we will notice that the architecture is emphasizing on generating an AI-based ACL rules (phase 3) depending on the alerts from the Intrusion Detection System IDS (phase 2). The analysist (phase 4) will be responsible for monitoring the results of the IDS (phase 2) and regulating the countermeasures (phase 5) when examining the system. Our architecture's components are presented in Figure 2.

A. Data Processing

In this step, the preparation of data will be managed to distinguish the data to two types of alerts: old and new alerts. The old alert refers to the alerts that are initially coming from the IDS; while the new alert refers to the alerts that is coming from IDS after applying AI to the managed ACL. In other words, the system will receive concerning alerts previously due to the fact of being IDS always analyzing the traffic and sending alerts accordingly. Therefore, the input data is a combination of both types of alerts (old and new).

B. Existing ACL

We mean by this a dataset of existing ACLs. These data sets will contain examples of input queries and descriptions along with their corresponding ACL rules. It is worth mentioning that these datasets should cover a wide range of scenarios to train the module effectively.

C. ACL Management Training

In here, the AI model will be adjusted to our processed dataset. The adjustment involves training our module on the ACL rules to make it more knowledgeable and better at generating relevant ACLs. Some machine learning approaches are needed to train the model at this stage.

D. Integration

This step is the result of the combination between the existing ACL and the ACL management training unit. The integration will be beneficial for well training the system to new rules and as a result adapting to newly upcoming permissions. This will also help in optimizing the system's countermeasures.

E. Network Equipment Deployer

The countermeasures (phase 5) that were generated by the alerts of the IDS will be shared with the results of the newly integrated AI-ACL rules. The deployment process will help in generating flexible ACL rules that will be able to deal up with changes that may occur to the network.

F. IDS

Intrusion Detection Systems will include analyzing patterns and behaviors within our system to identify abnormalities from the norm. It will generate alerts when detecting unusual or suspicious actions. These alerts are usually based on pattern recognition techniques but as within our system it will be enhanced with the machine learning approach [7]. In our system, the IDS will be generating alerts when it finds activities that fall outside the predefined threshold.

G. Countermeasure

This phase plays a key role in our case. They help in dealing with alerts that the network analyst handles in order to stop or reduce threats impact on the network. These actions include stopping malicious IP-address, changing firewall settings and letting network analysts know the possible threats to deal with them effectively.

VII. FEASIBILITY OF AI IN MANAGING ACL

We implemented another system to verify the feasibility of AI in generating effective ACLs based on processing IDS alerts and we named it Snort IDS Alert Analyzer. The Snort IDS alert Analyzer provided in Figure 3 demonstrates the significance of reliability of GPT (Generative Pre-trained

Transformer) in achieving promising results [18] when dealing with the components of the proposed system architecture that was discussed earlier. By ensuring that alerts are consistently generated, perceived, processed, analyzed and acted upon, the snort IDS alert analyzer contributes into the operations of the proposed system architecture in: IDS alerts handling, receiving, forwarding, reading, processing and ACL management. The consistency between snort IDS alert analyzer and the proposed system is initially seen in the output visualization in Figure 4. We used streamlit a python framework web application [17] and combined it with one of AI models to automate and enhance the performance of analyzing and responding to IDS alerts. This section examines the feasibility of using AI in the Snort's IDS Alerts and how effective it is in analyzing and generating ACLs. The test was conducted using GPT-2 model in its XL (Extra Large) variant, because large language models are often ideal for natural language processing tasks [18]. The other reason of deploying GPT-2 XL is because it has a larger model size and was proven for effectively handling longer text generation tasks. Although newer models have been developed with other advantages such as vastly larger parameter sizes, our initial testing shows that adopting GPT-2 XL is reliable and can assist in our specific use case [21]. Additionally, our code uses, the transformers library with the GPT-2 XL model, along with its tokenizer and language model aiming to perform a text generation output from snort's IDS alert input.

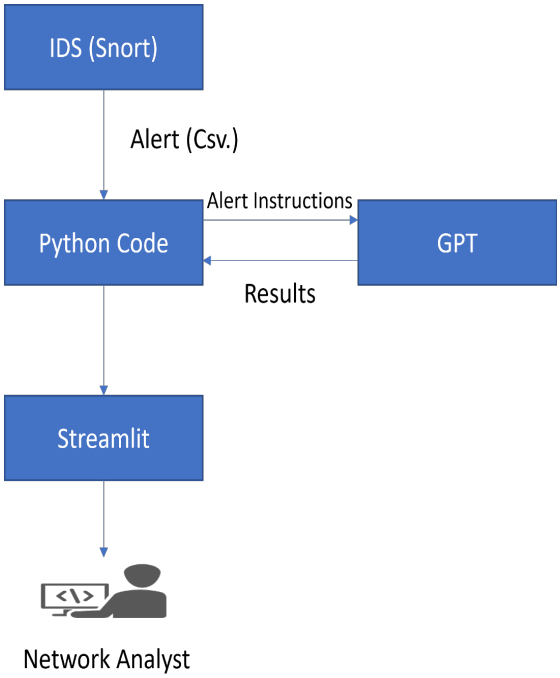


Figure 3. Snort IDS Alert Analyzer Design

Figure 3 demonstrates an organized task management of Snort IDS alerts. The workflow from alert reading to result displaying is closely controlled and firmly managed according to the modular task approach, promising a smooth handling of the task deployment [20]. The tasks are broken down into smaller tasks. Our approach enables the tasks to be divided into four instruction parts and then the python code incorporates these steps into a Streamlit app giving network analysts the chance to see and engage with the analysis and responses through an easy-to-use interface. These instructions are:

- Read Alerts: Getting snort IDS alerts from csv. file.
- Analyze Alerts: Identifying the severity level (high or low) for each alert and which ACL action rule will be associated with that alert.
- Generate Blocking Reasons: Providing a reason for blocking an IDS alert if GPT founded it suspicious.
- Generate Details: Displaying the alerts, the taken action and the blocking reasons.

The snort IDS alert analyzer mechanism relies on reading IDS alerts generated by Snort from a csv. file, and then analyses the severity and generates an ACL action response accordingly. If an alert that has some features such as indications of attack is considered critical, the system automatically blocks that IP address that is associated to the alert. AI will then recognize and analyze how severe the coming alert is and will generate a corresponded ACL rule.

- If the alert has a high severity with a deny ACL rule; AI will take an automated action by blocking the IP-address that is associated with that alert.
- If the alert has a low severity with an allow ACL rule; AI will let that connection pass without further action taken. The passed alerts will be manually reviewed by human analyst later if there is a need for further investigation.

Figure 4 shows the output when dealing with a Cross-site Scripting (XSS) alert and how does AI react to this alert. The reasons we believe this part of the implementation is feasible are primarily to its modular design and to the use of models beside streamlit [17]. GPT-2 is used on the explanation generation module for clear and contextually fluent reasons for the purpose of analyzing and taking security actions. It is clear that using AI (GPT-2 in this case) to automate the generation of ACLs from Snort IDS alerts is very promising.

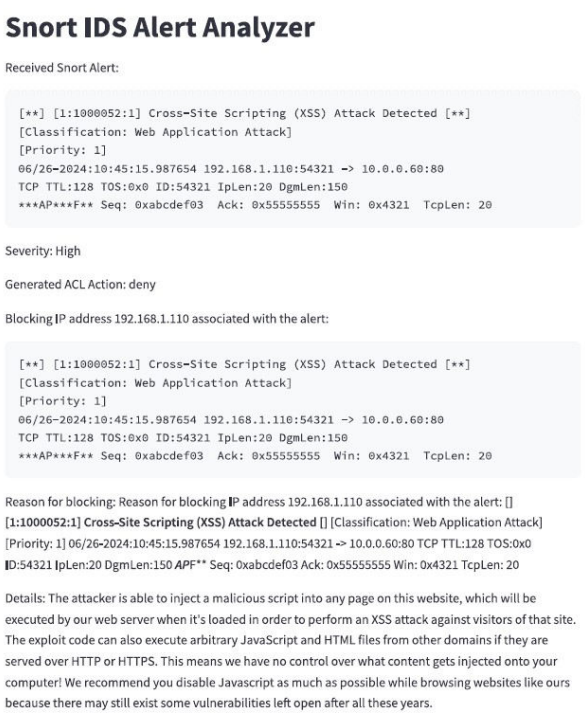


Figure 4. AI output example for ACL Management

The AI ability to deal up with complex sets of data, identify emergent patterns, and dynamically respond to changing threats can dramatically enhance network security operations, both in terms of speed and effectiveness. More tests are essential to evaluate the system in order to make it robust and versatile. Therefore, real network data should be used to test our system. The tested data should contain several IDS alert types and in various scenarios. By doing so, we can analyze the alerts behavior in different and more dynamic network environments. To check that, we can test it with as many alerts as we can, in order to optimize the processing techniques and the model's performance to work with a huge number of alerts at once (i.e., high-throughput scenarios). With all the tests we've run so far, we can also improve the interface, as well as add new features such as customized rules to handle what to do with each alert and other filtering options. Moreover, additional research should be conducted to boost performance, reliability and scalability, and to enable successful integration with the proposed system. This approach improves the efficiency of security operations, and infuses machine-learning insights into decisions to further enhance network defenses against the growing sophistication of cyber-attacks [19].

There are several assumptions to take into consideration when implementing our system. It is expected that our system will continuously fine-tune their behavior to match changing patterns and trends. In order to securely facilitate efficient anomaly detection and access control lists are primed to update based on new instances, and access-control policies are allowed to be finetuned over time.

Snort IDS alert analyzer uses the GPT-2 XL model and tokenizer from the Hugging Face Transformers library [21], and is designed to be run in a Streamlit environment (a library for developing interactive web-based interfaces) [17], which is a suitable environment to deal with a large number of alerts, where the alerts need to be processed in a more complex and a mission-critical style.

Furthermore, The IP-address format used in our system is IPv4, when the alert format changes by using IPv6 incorrect results will be generated as we have not tested this format in our case.

VIII. ASSUMPTIONS

Several assumptions have to be taken into consideration for a successful design, implementation, and analysis of the proposed system.

When implementing the system, we assume that computational resources will be available to be deployed for real-time alerts analyzing as well as to processing power. The system's performance results will be inaccurate if the available hardware is not properly maintained. We are emphasizing here about the need for continuous monitoring and probably upgrading the system's hardware so it will operate optimally when dealing with under heavy alerts load.

Another important assumption is regarding the accuracy and quality of the data that are fed into the system. The data here should be pre-processed and set according to the standards in the system. Inaccurate or distorted IDS format will lead to errors during the alert detection process, which will cause a manual interference to handle these errors effectively. We are stressing here on the importance of a well-defined information to ensure that the input data that enters the system is clean, structured, and free from noise that would impact the anomaly detection and access control management phases.

We assume that the AI model used in the proposed system will continuously learn and improve over time. The AI is therefore expected to treat the new alerts processed, as well as new countermeasures to be applied, as additional sources of education, reformulating its rules and ACLs dynamically in response to an evolving security threat landscape. This assumption is largely critical, as the network environment is always changing, new attack vectors come into the picture, and the system is to respond and readjust features in order to remain effective against emerging threats.

IX. CHALLENGES AND CONSIDERATIONS

The data we use in the Snort IDS Alert Analyzer shown in Figure 4 have a significant impact on how effective AI can be when implemented on the proposed system. In our case

we initially dealt with, GPT model [21], snort IDS alerts and ACLs as essential components to our system. However, to enhance the efficiency in our system; additional components are required. The dataset we used are considered to be sufficient but it should be extended and improved to get a better system performance. The needed data are in terms of continual, unbiased and contextually relevant anomaly detection and response creation [13]. Integration of these dataset will enable a much more refined and accurate analysis, which will lead to a greater resilience and reliability of the system as a whole.

False positives and false negatives are also possible [4]. Systems for detecting anomalies can produce false positives (which considered an indication of a network threat, when actually there is no threat exists), and false negatives (which it is an indication of no network threat, when actually there is a threat exists).

This puts greater demands on the complexity of the training datasets and, in turn, the sophistication of adversarial scenarios. This makes model development more difficult, and potentially expensive in terms of the computational resources required and the domain expertise needed to devise appropriate scenarios. To make the AI components such as GPT-2 more resilient to adversarial attacks, what is really important is to include adversary training to provide counter-examples during the training phase of the model; input validation ensures that the inputs are clean, e.g., they are in the expected format.

As most traffic communications are becoming encrypted to ensure user privacy and security, attackers have also started using encryption to mask their malicious activities, making it hard for security systems to detect and for network analysts to mitigate. Encrypted traffic does not follow the traditional analysis methods in inspecting the actual content of data packets as they appear obfuscated [33]. Analyzing encrypted traffic by the proposed system requires using additional mechanisms for identifying metadata and traffic patterns.

These difficulties and factors are highlighting the complexity in implementing access control lists, anomaly detection, and AI into one system architecture. Carefully addressing these issues will assist in creating a strong and reliable security framework.

X. Improving Access Control Lists with AI

As stated in our proposal, we can utilize AI to examine patterns and behavior to spot anomalies in network access requests. AI models used in the network systems can identify suspicious or suspicious access attempts and send notifications and take preventive measures by learning the typical behavior of users [8]. AI can be used to dynamically modify access control policies depending on current information and circumstances. AI algorithms are able to intelligently decide whether to give or refuse access in a more precise and context-aware manner by considering specific user behavior, device attributes, network information, and other related aspects.

ACL rules can be improved over time by AI algorithms that continuously learn from access patterns and security events. This adaptive learning strategy [8] can help in the evolution of ACLs to block unauthorized access more successfully while lowering false positives. AI algorithms can analyze large volumes of data related to user behavior, network traffic, and system logs to identify patterns, anomalies, and potential security risks [9]. This analysis helps in understanding the access requirements and potential threats [6], forming the basis for ACL generation.

Based on historical data and specified risk models, AI algorithms can evaluate the risk related to granting or rejecting particular rights. By taking into account elements like the user's role and potential vulnerabilities, AI models can provide access control policies that reduce security risks.

Network traffic, user behavior, and security events will be continuously monitored by AI, which may see changes and emerging patterns that can call for ACL adjustments [12]. By constantly modifying ACLs based on current findings, our proposed system can contribute to the maintenance of an efficient and up-to-date access control architecture.

AI classifies various kinds of network traffic and user behaviors using machine learning algorithms [11]. AI models can create ACL rules that permit or limit access based on particular categories or traits by comprehending these classifications.

XI. Discussion

Network traffic is monitored using Snort IDS in our system; AI can be used to increase the effectiveness in analyzing Snort IDS with GPT to give a realistic response and related explanation to these alerts [18]. By implementing our system, a network analyst not only can identify the abnormality but can also get information in a more meaningful and explanatory way, thus allowing for a better decision-making response.

GPT model is beneficial with the ACL-management in adding dynamicity to the access control process. The use of an AI-based component provides more intelligence to ACL management, where access can be requested and granted based on the context gathered from an AI-generated response to the alerts. This can help in IP flagging, analyzing, and mitigating suspicious IP-address through using ACL rules in a more context-aware and automated way. This is considered to be beyond the capabilities of conventional static rule-based ACL filtering approaches. The dynamic nature of network attacks means we will probably want to further train our AI models and update our IDS signatures as new patterns emerge.

Botnets have the ability to mimic authentic users [23] and can be detected using AI but it is not going to be an easy task to deal with. Due to the distributed nature of botnets and their complicated evasion tactics, it can be hard for AI

to detect the normal network traffic from abnormal ones. However, when using advanced techniques such as botnet fingerprinting in conjunction with proper machine learning models [24]; AI systems can better recognize these types of attacks. Moreover, integrating threat intelligence will also be important for a higher detection rate [25].

As security is deployed in our architecture (in the form of Intrusion Detection Systems, Access Control Lists and countermeasures), the architecture described above in Figure 2 can enhance zero trust networks [30]. Zero trust networks need to be authenticated and verified constantly to ensure the access requests are legitime [26]. In this context; the AI-generated ACLs will dynamically adjust access control policies in real time, based on the latest threats and network activity. This means the network is always protected based on up-to-date insights. Continuously analyzing old and new alerts allows the system to put in place the newly generated ACLs aligned with zero-trust principles. Even internal actors must be verified in a zero-trust network. This will ensure the integrity of people and resources that can bypass security controls through the automatic generation and deployment of ACLs based on old and current alerts.

Our system needs enhancements to distinguish between legitimate and malicious behaviors to minimize the effect of attackers who try to resemble the patterns of legitimate users. When such attackers imitate legitimate user actions, their behaviors can still be exposed and detected through deep alert analysis [28][29]. Behavior analysis can also play a significant role here as it shows an effective way to differentiate between legitimate users and malicious actors [31]. The proposed system by the help of AI, has to continuously learn about the typical behavior of authorized users, and needs to point out any suspicious deviations by using applicable machine learning approaches [32]. Moreover, the system must adopt machine learning models to analyze metadata and identify suspicious traffic with much better efficiency [34]. This will lead to group the investigated traffic to a manageable set for detailed examination performed by network analysts.

Integrating machine learning into this system brings promising benefits with the rise in traffic encryption [35]. Machine learning enhances the functions of IDS, as the system learns to adapt continuously to new patterns and threats. AI and countermeasure processes will now be dynamic and automated in nature, as machine learning optimizes ACL generation, deploys countermeasures, as well as to network performance. This will enhance the network analyst's role with ML-driven insights, making the approach to network security proactive and dynamic.

XII. CONCLUSION AND FUTURE WORK

In this paper, an architecture to manage ACLs to detect the suspicious traffic and to thus to secure our network was presented. It will help security analysts to make wise decisions based on the results that AI capabilities bring by identifying patterns and predicting potential threats proactively. This work introduces the pros and cons of managing ACL using AI in security systems. In our future work, we will look into adapting ACLs based on anomalies and policies. Adaptations will provide 'digital immunity' to quickly perform the required containment and mitigation actions upon the detection of a threat. Adaptations can also help improve the resilience of network defenses against evolving threats. We would also be making the system more reliable by continuing to adjust AI models and examine the integration of other cybersecurity concepts. System modifications will change the behavior pattern of a system. In this case, AI must adapt to recognize the new changes in order to avoid misclassifying legitimate changes as suspicious or even mistakenly considered to be attacks. Therefore, network administrators need to guarantee the performance of the system after such updates are conducted. This will include; the type of the update, the type of IDS alerts and the applied corresponded ACL. Fine-tuning the required parts of the system will take hours to days and the retraining process with new data will take weeks. Therefore, real-time monitoring, periodic retraining and considered adjustments can ensure the system will employ the processes needed for ACL to be Generated accurately. One of the open aspects in this research is the possibility of attackers exploiting the system by inducing false positives on the IDS, which could lead to DoS attacks. Although this concern has been identified, no thorough evaluation of the system's resilience to such attacks has been conducted. As future work, we will analyze this vulnerability by finding ways in which the system could be secured against such malicious users, with a view to ensuring that false positives do not impact on either the availability or the reliability of the network. We hope that such an extensible and effective system will help network analysts quickly respond and resolve more and more alerts when faced with new and evolving threats. Our goal is to support critical infrastructures and the integrity of data in complex threat environments. It exhibits great advantages as it enables adaptive threat detection and response and can be a solution to security threats that are constantly evolving. Accordingly, addressing these challenges poses numerous opportunities to achieve better protection of our network from security threats.

ACKNOWLEDGMENT

This work was partially supported by JSPS KAKENHI Grant Number JP19K20268, JP24K14959.

REFERENCES

[1] N.Shahata, H.Hasegawa, and H.Takakura, "AI-driven Approach for Access Control List Management," Proc. of The Seventeenth International Conference on Emerging Security Information, Systems and Technologies, 2023, pp. 52 – 58.

[2] N. Muhammad, U. Shams, B. Mohammad, "Network intrusion prevention by configuring ACLs on the routers, based on snort IDS alerts," *IEEE Communications Surveys & Tutorials,* vol. 22, no.2, pp. 1392-1431, Oct. 2010.
[3] C. Lee, J. Kim and S. Kang, "Semi-supervised Anomaly Detection with Reinforcement Learning," *Computers and Communications (ITC-CSCC),* Phuket, 2022, pp. 933-936, Jul. 2022.
[4] C. Varun, B. Arindam, and K. Vipin, "Anomaly detection: A survey," *ACM Computing Surveys,* vol.41, no.3, pp. 1-58, Jul. 2009.
[5] C. Raghavendra, and C. Sanjay, "Deep learning for anomaly detection: A survey*". arXiv:1901.03407*, Jan. 2019.
[6] C. Kukjin, Y. Jihun, P. Changhwa, and Y. Sungroh, "Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines," *IEEE Access*, vol. 9, pp. 120043 – 120065, Aug. 2021.
[7] H. Victoria, and A. Jim, "A Survey of Outlier Detection Methodologies," *Artificial Intelligence Review*, vol. 22, Springer, pp. 85-126, Oct. 2004.
[8] B. Anna, and G. Erhan, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153 – 1176, Oct. 2015.
[9] H. Yassine, G. Khalida, A. Abdullah, B. Faycal, and A. Abbes, "Artificial intelligence-based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives," *Applied Energy*, vol. 287, pp. 1-26, Apr. 2021.
[10] Z. Shuai, C. Mayanka, L. Yugyung, and M. Deep, "Real-Time Network Anomaly Detection System Using Machine Learning,"*IEEE*, pp. 267-270, Jul. 2015.
[11] D. Kyle, H. Abdeltawab, and A. Marco, "A Survey of AI-Based Anomaly Detection in IoT and Sensor Networks," Sensors*,* vol. 23, no. 3, Jan. 2023.
[12] L. Xiao, H. Brett, and W. Dinghao, "Automated Synthesis of Access Control Lists*,"* Proc. International *Conference on Software Security and Assurance (ICSSA)*, Altoona, pp. 104-109, Jul. 2017.
[13] Z. Shakila, A. Khaled, A. Mohammed A, A. Muhammad Raisuddin, K. Risala. Tasin., K. M. Shamim, M. Mahmud, "Security Threats and Artificial Intelligence Based Countermeasures for Internet of Things Networks: A Comprehensive Survey," IEEE *Access*, vol. 9, pp. 94668-94690, Jun. 2021.
[14] Twingate, "Access Control Lists (ACLs): How They Work & Best Practices". [Online]. Available from: https:twingate.com/blog/access-control-list/ 2023.07.25
[15] Dandelife, "Understanding the Pros and Cons of Access Control Lists". [Online]. Available from: https://dandelife.com/understanding-the-pros-and-cons-of-access-control-lists/ 2023.07.26
[16] I. Muhammad, W. Lei, M. Gabriel-Miro, A. Aamir, S. Nadir, M. K. Razzaq, "PrePass-Flow: A Machine Learning based technique to minimize ACL policy violation due to links failure in hybrid SDN," *Computer Networks*, vol. 184,107706, Jan. 2021.
[17] Streamlit, [Online]. Available from https://streamlit.io/ 2024.03.20
[18] Ali.T, "Next-Generation Intrusion Detection Systems with LLMS: Real-time Anomaly Detection, Explainable AI, and Adaptive Data Generation," pp. 1-65.
[19] R. Trifonov, O. Nakov and V. Mladenov, "Artificial Intelligence in Cyber Threats Intelligence," *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)*, Mon Tresor, 2018, pp. 1-4.
[20] P. K. Mannam, "Optimizing Software Release Management with GPT-Enabled Log Anomaly Detection," *2023 26th International Conference on Computer and Information Technology (ICCIT)*, Cox's Bazar, 2023, pp. 1-6.
[21] Huggingface [Online]. Available from https://huggingface.co/ 2024.03.25
[22] J. Lansky et al., "Deep Learning-Based Intrusion Detection Systems: A Systematic Review," *IEEE Access*, vol. 9, pp. 101574-101599, 2021.
[23] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, "Design and analysis of a social botnet," *Computer Networks*,Vol. 57, no. 2, pp. 556-578, 2013.
[24] R. Vijayakanthan, K. M. Waguespack, I. Ahmed and A. Ali-Gombe, "Fortifying IoT Devices: AI-Driven Intrusion Detection via Memory-Encoded Audio Signals," *2023 IEEE Secure Development Conference (SecDev)*, Atlanta, GA, USA, 2023, pp. 106-117.
[25] R. Trifonov, O. Nakov and V. Mladenov, "Artificial Intelligence in Cyber Threats Intelligence," *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)*, Mon Tresor, Mauritius, 2018, pp. 1-4.
[26] P. Assunção, "A zero trust approach to network security," Proc. the Digital Privacy and Security Conference, vol. 2019. Porto Portugal, 2019, pp. 65-72.
[27] L. Yee Por *et al*., "A Systematic Literature Review on AI-Based Methods and Challenges in Detecting Zero-Day Attacks," *IEEE Access*, vol. 12, pp. 144150-144163.
[28] S. McElwee, J. Heaton, J. Fraley and J. Cannady, "Deep learning for prioritizing and responding to intrusion detection alerts," *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*, 2017, pp. 1-5.
[29] A. Imeri and O. Rysavy, "Deep learning for predictive alerting and cyber-attack mitigation," *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC),* Las Vegas, 2023, pp. 0476-0481.
[30] D. Eidle, S. Y. Ni, C. DeCusatis and A. Sager, "Autonomic security for zero trust networks," *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, New York, 2017, pp. 288-293.
[31] W. Zhenqi and W. Xinyu, "NetFlow Based Intrusion Detection System," *2008 International Conference on MultiMedia and Information Technology*, Three Gorges, China, 2008, pp. 825-828.
[32] S. Saad et al., "Detecting P2P botnets through network behavior analysis and machine learning," *2011 Ninth Annual International Conference on Privacy, Security and Trust, Montreal*, 2011, pp. 174-180.
[33] Natureprotfolio. *Vigilance still critical in highly encrypted networks.* [Online]. Available from: https://www.nature.com/articles/d42473-023-00326-y/ 2024.11.22
[34] S. Hiruta, I. Hosomi, H. Hasegawa, and H. Takakura, "Security Operation Support by Estimating Cyber Attacks Without Traffic Decryption," *Proc. 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, Torino, 2023, pp. 1127-1132.
[35] M. Shen et al., "Machine Learning-Powered Encrypted Network Traffic Analysis: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 25, pp. 791-824, 2023.

Science-Tracking Fingerprint: Track the Tracker on the Example of Online Public Access Catalogs (OPAC)

1st Stefan Kiltz
Faculty of Computer Science
Otto-von-Guericke-University of Magdeburg
Magdeburg, Germany
stefan.kiltz@iti.cs.uni-magdeburg.de

2nd Nick Weiler
Faculty of Computer Science
Otto-von-Guericke-University of Magdeburg
Magdeburg, Germany
nick.weiler@st.ovgu.de

3rd Till-Frederik Riechard
Faculty of Computer Science
Otto-von-Guericke-University of Magdeburg
Magdeburg, Germany
riechard@ovgu.de

4th Robert Altschaffel
Faculty of Computer Science
Otto-von-Guericke-University of Magdeburg
Magdeburg, Germany
robert.altschaffel@iti.cs.uni-magdeburg.de

5th Jana Dittmann
Faculty of Computer Science
Otto-von-Guericke-University of Magdeburg
Magdeburg, Germany
jana.dittmann@iti.cs.uni-magdeburg.de

Abstract*—We are motivated by the Science-Tracking Fingerprint (STF) from our companion conference article 'Science-Tracker Fingerprinting with Uncertainty: Selected Common Characteristics of Publishers from Network to Application Trackers on the Example of Web, App and Email' and apply this fingerprint concept to Online Public Access Catalogs (OPAC) provided by many libraries for literature research with the aim to track the tracker. We choose an approach rooted in digital forensics and using only open source, on-premises tools for comprehensibility and repeatability purposes. The goal of this article (together with its companion article) is not primarily to detect the amount of tracking that is taking place. Studies towards that goal have, indeed, been conducted both on the Science-tracking field and on the field of tracking in general. Our goal is to try and identify the publisher based on the employed first and third party tracking. In particular, for the application area of web we enhance the concept from the companion article with an automated acquisition, investigation and analysis process, including the calculation of the STF. Further, the single list of trackers from the companion article is extended and we provide 3 different lists of known trackers in order to increase the hit-ratio for known tracker domains. For the automation we introduce a toolset consisting of 6 self-created software tools and 4 automation scripts that are put into open source. The automation enables a substantially larger investigation on both the tracking habits of publishers and allows evaluations of the stability of the Science-Tracking Fingerprint. In total we fully evaluate 60 downloads from the 4 exemplary chosen individual publishers across 3 different test-series. Further, to detect any possible influence of the changes of the domains contained in the tracker lists, we use 3 different versions of each of the 3 tracking lists and apply it to each test series. The results of our in-depth study into Science-Trackers show that some publishers change their embedded trackers over individual papers and articles (intra-publisher diversity). For the duration of the tests, no changes on the content of the tracking lists relevant to the tests occurred. Results from 4 tested publishers show no difference in the observed tracking between open access and non-open access articles. Further, we show that using the exemplary chosen OPAC instance of our university library does not prevent Science-Tracking by the publishers, potentially contrary to the user's expectations. This article proposes a comprehensible, scientific process to support the identification of the tracking party (publisher) based on the trackers employed by the tracking party.

***Keywords*-Security, trust and privacy metrics; IT forensics; Attribution.**

The research from Stefan Kiltz, Robert Altschaffel and Jana Dittmann is partly funded by the EFRE project "CyberSecurity-Verbund LSA II – Prävention, Detektion und Reaktion mit Open Source-Perspektiven" https://forschung-sachsen-anhalt.de/project/cybersecurity-verbund-lsa-praevention-27322.

I. Introduction

Science-Tracking by publishers, as stated in [1] is in widespread use ([2], [3]). This often stealthy practice subjects users of literature information systems to unwanted data processing and impacts their privacy, sometimes with potentially grave consequences [2]. On a side note, data from scientists can also be obtained and sold through breaches in conference registration systems etc. (see e.g., [4]).

To get an overview of the extent of the tracking of scientists by publishers, an IT-forensic approach as motivated in [5], conforming with [1] can be a valid course of action. As already pointed out in the companion conference article [1], each forensic investigation method comes with the potential for error, loss and uncertainty, which can influence the resulting traces. Hence, using results from multiple, independent tools for the same forensic research goal is used to reduce these negative effects. Further, our goal is to gather hints/leads

leading towards an individualization of a publisher based on the trackers employed (first and third party)

The ability to identify instances of tracking open the way to investigate interesting questions about the extent and practical use of tracking. This work aims to answer the following primary research questions (***RQ***), extending the companion conference article [1]:

- ***RQ1*** *Track the tracker:* whether it is possible to find traces/hints that allow for an individualization (attribution) of the publisher that employs the tracking mechanisms
- ***RQ2*** *Intra-publisher diversity:* how stable the traces are over time for a given publisher and within multiple documents from the same publisher
- ***RQ3*** *Countermeasures against tracking by using OPAC:* whether the usage of library-supplied research gateways such as an Online Public Access Catalogues (OPAC) prevents the tracking techniques employed by the publishers
- ***RQ4*** *Effect of open-access on tracking:* whether there is any noticeable difference in the tracking behaviour when accessing open-access and non-open access articles.

Based on [1] we trade broadness for detail in our research and focus on Science-Tracking on the example application area of web-based services accessed via browser. We choose a scenario that reflects the typical usage of scientific literature research using our university library and the Online Public Access Catalogue (OPAC) gateway [6] used therein. This is particularly interesting since users could expect to fetch the documents proxied by the university library and this could lead them to suspect that they are not tracked by the publishers in the same way as stated in [1]. We contribute a semi-automatic approach to calculate the Science-Tracking Fingerprint (STF) for the web application area. With the partially automated support, we can look into changes in detected tracking mechanisms per publisher using different articles and different points in time (intra-publisher diversity).

Addressing these research questions includes various steps, concepts, extensions and improvements over the companion publication [1] that might also be applied to other research questions in the future. These are:

- ***Extension E1*** - an inter-publisher comparison of intersected STFs for estimating the difference in tracking behaviour between publishers.
- ***Extension E2*** - the creation of a STF-deviation metric to show the difference between different STFs.

These extensions are necessary to identify the various tracking parties and hence address ***RQ1***.

- ***Extension E3*** - the concept of the evaluation of tracking across multiple documents and points in time (t_i, t_{i+1} see [1]) for an individual publisher (intra-publisher diversity) and its comparison using the STF.

The extension ***E3*** is necessary to investigate the diversity of tracking methods used by a specific publisher (***RQ2***).

Furthermore, the extension of the system landscape is necessary in order to investigate ***RQ3***, which is related to OPAC and hence requires its inclusion.

- ***Extension E4*** - the inclusion of an Online Public Access Catalog (OPAC) library gateway to the publisher's articles into the system landscape used for research, which adds a credible scenario of Science-Tracking in common literature research.

Additional noteworthy extensions to the work performed in [1] are provided in the following. They either extend previous work, simplify future forensic investigations or cover notable findings:

- ***Extension E5*** - a partially automated process consisting of 10 scripts that are put into Open Source and covering the acquisition, investigation and partly analysis according to the sets of investigation steps from [7] for calculating the STF.
- ***Extension E6*** - the usage of multiple lists of known trackers for the investigation, saved at different points of time, to have higher chances of detecting trackers and their analysing tracking detection behaviour.
- ***Extension E7*** - an additional analysis of the publisher Wiley to broaden our group of investigated publisher.
- ***Extension E8*** - the discovery of different tracking behaviour depending on the type of browser (interactive vs. headless with automated control flows.

With both implementing an automated process within the investigation (***E5***) and with the concept of the evaluation of tracking across multiple documents and points in time for an individual publisher (***RQ2***) and its comparison using the STF (***RQ1***) we are addressing the future work suggested in the companion conference article [1].

This article is structured as follows: In Section II aspects of the relevant state of the art are outlined briefly. Section III describes the necessary fundamentals for understanding the concept, implementation and evaluation of this article. In Section IV we discuss our conceptual approach centred around a model of the forensic process and introduce the STF-deviation as a metric to describe differences between STFs. In Section V we describe the implementation of the concept using pseudo-code to illustrate the workings of the 6 self-created software tools. In Section VI the concept and its implementation is evaluated, forming the contributions outlined in Section I. This article closes with a conclusion and an outlook regarding future work in Section VII.

II. State of the Art

As already stated in [1] a number of studies exist that look into data tracking in general. For instance, Wolfie Christl in [8] investigates digital tracking and profiling by corporate networks and their implications for the user ranging from individuals to society at large. On the technical side the research covers the practices of recording, combining, sharing, and trading of personal data. The main effort is directed at

mapping of today's personal data ecosystem and determining its scope. The study from Mildebrath ([9]) takes a detailed look on the tracking mechanisms and practices employed by Google, Facebook and Amazon, both on the web and using mobile app infrastructures. The focus of the study from Samarasinghe et al. [10] is put on the influence of the geolocation of a tracked user by differentiating the tracking results from 56 countries based on a selection of frequently accessed websites. The study from Sim et al. [11] primarily focuses on existing tools and measures to detect (tracking-measurement) and prevent various types of web-based tracking and also glances into app-based tracking. Also addressing prevention of tracking, the study from Pan et al. [12] looks at the success of the attempt of browser manufacturers to block tracking mechanisms. The measurement of the success is performed using available privacy scanner and its conclusion is a slight reduction of tracking by modern browsers on the example of Google Chrome. Geared towards the field of mobile devices, the study from Krupp et al. [13] focuses on mobile devices, which offer lesser tracking protection based on the fact that privacy enhancing browser add-ons and extensions are typically unavailable for the apps. The research focuses on iOS devices and reveals a substantive amount of tracking in the apps chosen for the research by the authors.

Science-Tracking, which is the subject of this article and its companion conference article [1], can be looked upon from very different angles, e.g., primarily from a legal perspective as conducted in the article from Altschaffel et al. [14]. The study done by Hanson [15] looks into the extent of Science-Tracking from a technical perspective. Key findings also include the huge amount of third party tracking by third-party code being loaded whilst accessing an article's page provided by a publisher. The tracking mechanism provided by the third parties employed by the publishers identified by [15] seem to primarily consist of the generic third party tracking solutions also employed in general tracking as outlined in the above paragraph, which also mirrors our findings from [1].

All reviewed studies share the fact that they try to determine to what extent tracking exist on various application fields and elaborate on the consequences of user tracking. The study [5] already employs forensic techniques for the detection of tracking. According to our knowledge, [1] is the first attempt at a study with forensically motivated systematic means to give hints/leads to individualize (attribute) tracking to identify an originator. Hence, this publication is used a foundation for our work. In this article at hand the approach outlined in Section IV is a refined attempt at fingerprinting originators of Science-Tracking (organizations such as publishers) on the basis of their employed first and third party tracking mechanisms also for the task of comparing different originators. The authors are fully aware that the suggested approach alone will not suffice for individualization and thus attribution but believe that it can give hints/leads towards further investigation.

The topic of data tracking is also of interest outside the field of academia. For instance, the European Union *Study on the impact of recent developments in digital advertising on privacy, publishers and advertisers* [16] investigates the tracking of users a foundation for targeted digital advertisement. The study investigates the data reported and the means employed by publishers to do so. As such, it provides a broad understanding of data tracking but does not provide any means to measure the occurrence of data tracking.

III. Fundamentals

This section describes the necessary fundamentals for the research presented in this article. It relies heavily on the findings, fundamentals and findings from our companion conference article [1].

A. The Data-Centric Examination Approach (DCEA) forensic process model

A comprehensive, model-based approach (as also used in [1] supports the forensic soundness. The Data-Centric Examination Approach (DCEA) [7] uses data streams and forensic data types, which together with forensic methods (represented by capabilities of forensic tools) supports a detailed description of the provenance of the data from the beginning of the examination to its end. This is seen by the authors as an aid to attribution. The model from [7] distinguishes three data streams:

- Mass storage data stream DS_T (time-discrete, low volatility, long-term data retention),
- Main memory data stream DS_M (time-discrete, high volatility, short-term data retention),
- Network data stream DS_N (time-continuous, high volatility, short-term data retention).

Throughout this article (as in [1]) we will use DS_T and DS_N during our examinations. Those data streams can be further divided into 8 forensic data types with the assumption that data of a specific data type is created, processed, stored and used similarly by a given IT system and thus can be acquired, investigated, analyzed and documented similarly in a forensic examination [7]. For our article (as in [1]) we use DT_3 (details about data) and DT_5 (communication protocol data) in the context of the network data stream and its representation in mass storage.

The collection of main memory data from DS_M and its examination, whilst being available in theory, is omitted due to the extra effort weighted against the additional information gained. It would involve halting the VM used for the examination to capture the RAM content for each point of interest during the examination and creating a dwarf specifically for the examination environment with the Volatility framework (see e.g., [17]) and browsing the processes for relevant data. The authors believe that capturing highly volatile data in the shape of DS_N and DS_T for data with low volatility represents a measured approach and a good balance between effort and the gain with regards to data containing relevant information for the research.

The system landscape analysis is, according to [7], part of a forensic examination. The spatial and temporal intricacies

of tool placement and operation define what can be obtained and analyzed. As stated in [5], the usage of on-premises tools allows for finer control over the tool operation and external data (e.g., lists used for comparison against known tracker URLs) and better data access (e.g., regarding intermediate results). Opposed to the original work in [1] we will use exclusively on-premises tools and rely on corroboration of the tool results of the different on-premises tools. This enables a finer control over the tool configurations and external data used. In Section IV-D we discuss the properties of both approaches with our system landscape analysis.
The existing model-based approach of the forensic examination as described in [7] alone is not sufficient for the individualization (attribution). However, it provides us with the elementary building blocks for the fingerprint (e.g., data streams, forensic data types).

B. Selected tools and data sources for URL and Tracker examination

We select existing tools based on their proven functionality (analogous to the companion conference article [1]) and combine them in scripts (bash- and python-based) that cover different tasks of the investigation process. The choice of tools is based on the following requirements:

- Open Source: the tool must be comprehensible and potential changes on the source code must be possible
- Maintenance: the code must be maintained and updated by the tool authors
- On-premises installation: access to the data collected (including intermediate data and examined must be strictly local
- Forensic operation: the tool must not alter the immediate data nor alter the behaviour of the client or server software

Frameworks such as OpenWPM [18], whilst being generally suited for privacy measurements, can violate some of the requirements (e.g., due to using the Firefox engine, which can automatically start connections unrelated to the measurements such as software and certificate updates, contacting safebrowsing service providers etc., interfering with the data in the network data stream DS_N). We further select tools such as Webbkoll [19], although they only collect a subset of data of privacy measurement tools, based on the goal of our research regarding individualization of publishers based on their employed first and third party tracking mechanisms. Using the terminology from [1] we use tools that operate both statically and dynamically. The forensic data types and data streams (see Section III-A) are used from [7]. Contrary to [1] we only use on-premises tools for full source-level control over their functionality and parameterization. The existing tools used in the scripts combined are:

- Webbkoll [19]: on-premises, operating on the network data stream DS_N on Raw Data DT_1 and yielding tracker output DT_3 (in conjunction with external data, i.e., tracker list data) as results as well as URL and IP data DT_5 as results, both output to the mass storage data stream DS_T
- TShark [20]: on-premises, operating on the network data stream DS_N on Raw Data DT_1 and yielding URL and IP data DT_5 as results on the mass storage data stream DS_T
- Website evidence collector [21]: on-premises, operating on the network data stream DS_N on Raw Data DT_1 and yielding URL data DT_5 as results on the mass storage data stream DS_T

The tools used in our research (see Section III-B) utilize a headless version (without graphical interface) of the chromium browser [22]. For this the library Puppeteer [23] is employed to provide an easy as well as time and resource efficient way of implementing the forensic tools in a headless environment. The data sources for the 60 papers originate from our university's library OPAC gateway that are redirected to the 4 selected publishers:

- Association for Computing Machinery ACM Inc.
- Elsevier
- Institute of Electrical and Electronics Engineers IEEE
- Springer Nature

All recordings are conducted at the dates of:

- 20/02/2024
- 12/03/2024
- 25/03/2024

For the external data we use the sources of the lists of:

- Disconnect [24]
- Easy Privacy [25]
- Fanboy Annoyance [25]

These lists provide the classification of a given domain as a tracker. They are used for the dynamic examination (see also [1]) of the recordings created by TShark. A decision is reached whether a given domain is a tracker by comparing them against the lists. Different lists are used to render the results more plausible. We acquire the list data at the dates of:

- 20/02/2024
- 25/02/2024
- 13/03/2024

With those differently timed versions of the lists, we can conduct experiments regarding changes in detection depending on the changing content of the 3 lists over time. With this setup we can address the point raised in [1], which at a minimum asked for the dates to be recorded alongside with the result for comparability. Our setup allows for retrospective runs of the tests on the data with arbitrary dated lists.

C. Uncertainty in forensic examinations

Uncertainty is a property that should be factored in for all forensic examinations [1]. This is laid out in detail in [26]. For the approach in [1], which is adapted for usage in the research described in this article, the certainty category therein is also employed. This certainty category from [1] weighs the results of different forensic tools capturing URL-data and

tracker detection data as matches of the results being plausible, uncertain or non-existent, depending on whether all tools agree with the results, at least one tool returns a diverging result or no matches exist at all.

D. Semantics and syntax of the Science-Tracking Fingerprint (STF)

In the companion conference article [1] the semantics and syntax of the Science-Tracking Fingerprint (STF) are introduced. Here, we provide a brief summary of the concept as a basis for this article.

One goal of the STF is the support for individualization [27] and attribution of the publisher employing the tracking techniques (track the tracker). The general idea, with regards to the semantics of the STF, is to employ more than one forensic method to acquire, investigate and analyse the data in the absence of a ground truth when accessing the articles supplied by the publisher. We record the agreement (matches) of the respective tool results according to the certainty categories (see Section III-C) of:

- plausible (pl): all tools return the same or comparable result,
- uncertain (unc): at least one tool returns a diverging result,
- none (-): no tool returns a meaningful result.

Semantically, the Science-Tracking Fingerprint can be described as a matrix of A-Records for first and third party as well as CNAME domain names for first and third party on one axis and Web, App and Email on the axis. Each cell contains a structured description covering the following elements:

- Counter: Number of occurrences,
- Certainty: plausible, uncertain or none,
- Data stream: Mass storage (T) or Network (N),
- Data type: DT_5 (URL) or DT_3 (Tracker),
- Discovery mode: list-based (L) and/or manual (M).

A fixed structure for the notation of these elements is necessary to support comparisons between the findings obtained with different forensic methods. The structured description is summarized:

```
1 <CELL> ::= <Counter> <EXPR>
2 <EXPR> ::= <EXPR1> | <EXPR>,<EXPR1>
3 <EXPR1> ::= <Certainty>,<Data stream>,<Data type> |
4 <Certainty>,<Data stream>,<Data type>,<Discovery
      mode>
```

Listing 1. Structured description for the cell contents formed from relevant elements.

The semantics of the STF describe quantifiable and qualitative differences between the Science-tracking employed by the publishers, with changes over time to be expected, which is why the STF is treated as a similarity measure [1].

According to [1], the syntax of the STF can be described a concatenation of vectors consisting of element value pairs, which form the matrix shown in Figure 1.

	A-Record 1st Party	CNAME 1st Party	A-Record 3rd Party	CNAME 3rd Party
Web	<CELL>	<CELL>	<CELL>	<CELL>
	<CELL>	<CELL>	<CELL>	<CELL>
App	<CELL>	<CELL>	<CELL>	<CELL>
	<CELL>	<CELL>	<CELL>	<CELL>
Email	<CELL>	<CELL>	<CELL>	<CELL>
	<CELL>	<CELL>	<CELL>	<CELL>

Figure 1. Syntactical matrix representation of the STF according to [1].

Each row (according to [1]) consists of a set of cells that are ordered according to the URL specifics (DT_5), namely the A-Record and CNAME domain name entries for both first and third party, respectively. Those cells can also be empty (represented by a 0), if there are no domains in the investigated recording. The part of the counter in the cell describes numbers of occurrences according to the following conditions:

- matching certainty per cell,
- tracker certainty is either plausible or uncertain.

A special case is met when a row contains entries where the DNS response provided URL information containing CNAMEs for the first and/or third party. In [1] it is described to duplicate the cell entries from the A-Record to the CNAME without increasing the counter value, as this case with CNAMEs first and/or third party in one row technically describes the same examination step.

As stated in Section I, in this article we are only using the Web application area part of the syntactical representation of the STF.

IV. Conceptual Approach

This section describes the conceptual approach to the web-based investigation performed in this article. The approach focuses on collecting DT_5 and DT_3 data from as many publications as possible (to reduce the potential error, loss and uncertainty, see Section I) by intersecting the sets of gathered trackers from different publications of a publisher. The investigation is performed on a test series. It uses the set of examination steps from [7] (see Section III-A). We discuss in detail the three steps of:

- Data gathering
- Data investigation
  - Generation of result tables and STFs
  - Aggregation of STFs
- Data analysis

The complete analysis process is shown in Figure 2. It outlines the three main steps (data gathering, data investigation, data analysis), the input (test set, external list data, see Section III-B) and intermediate results and the analysis questions to be answered.

Data gathering in essence marks the acquisition of data. It only gains raw data DT_1 for further investigation and analysis in the following steps.
By including the Online Public Access Catalog (OPAC) gateway provided by our universities library this puts restrictions on the location of the acquisition device (see Section IV-D) but allows us to see the perspective of the researchers using the library services (see ***Research Question RQ3*** by employing ***Extension E4*** in Section I).
By using different types of browsers (interactive vs. headless) during data gathering we extend the research from the companion article [1] and provide the ***Extension E8*** (see Section I).
The selection of the types of documents (open-access vs. non-open-access) to be queried during data gathering addresses the ***Research Question RQ4*** (see Section I).
The process of the generation of result tables and STFs as part of the data investigation step allows for multiple comparisons against different versions of the tracker lists from the documents already gathered enhances the findings from the companion article [1] as the ***Extension E6*** (see Section I).
Intersecting the generated result tables and STFs provides further insight into intra-publisher diversity and inter-publisher differences addresses the ***Research Question RQ2*** and enhances the findings from the companion article [1] as the ***Extension E3*** (see Section I).
The general design of the examination process with a focus on automation enhances the findings from [1] as the ***Extension E5*** (see Section I) while adhering to the model from [7]. It thus ensures a correct re-iteration of each step, which enhances the findings from [1] as the ***Extension E3*** (see Section I).
In the following, we will describe details regarding each of those selected examination steps.

A. Data gathering

During the acquisition, the necessary data is collected via the described tools in Section III-B and saved to the mass storage. While Webbkoll [28] and Website Evidence Collector initially gather raw data DT_1 internally for later investigation of the website for possible third party hosts, TShark records the network traffic as raw data DT_1 (for later external investigation and analysis) whilst querying the publisher website for the literature. The acquisition must be performed within the network of the university ; without an explicit login access to the papers provided by the OPAC of the university is impossible. Further processing of the gathered data may be performed elsewhere.
Interestingly, the choice of the type of browser, headless or graphical browser, influences the recording of the network data (see Section VI-A) and forms our ***Extension E8*** in Section I). Although at first counter-intuitive since researchers use a graphical browser in their daily research, we choose to use headless browsers on the grounds that:

a) this is also used in commonly accepted forensic tools such as Website Evidence Collector [21],
b) because it allows for automation and thus enables an examination for a much larger figure of documents.

To get more insight into the influence of the used type of browser on tracking behaviour, sample recordings with a graphical browser are conducted to compare the amounts of gathered data in both cases. The findings of this sample to our research data are detailed in Section VI-A.

B. Data investigation

We highlight the two steps that are performed during data investigation step that is following the data gathering step. The data investigation is partial automated by using self-created scripts.

1) Generation of result tables and STFs: With the collected data from the publisher websites (in our case of our research totalling 2.1GB, see also Section V-A for technical data on the devices used), a result table listing all discovered third party hosts is generated based on [1].
For this, the relevant DT_5 data:

- host name,
- ip address,
- whether host is third party,
- host is A Record or CNAME (see [1] and Section VI-A),

is gathered from the output data and combined to a structure. This structure is checked, by identifying whether a host is known in a list or not, gaining DT_3 data. This check is performed with every list and the result of each check is kept separately, since there could be differences within the lists. Once all hosts were checked on each list, a DT_3 and DT_5 match will be performed to grade the plausibility of the detected tracker. If on either DT_3 or DT_5 match at least one result of "uncertain" was achieved, the host is classified as a potential tracker [1]. After all checks have been performed on each gathered host, the result table and STF are generated based on the information gained. To also cover the possibility of change in detection of trackers over time, each tracker list has a version related to the date of data acquisition. In Listings 3 and 4 from Section VI-A the pseudo-algorithmic approach of the evaluation and the update of the STF for every host is shown.

2) Aggregation of STFs: When the test series is processed completely, an aggregation based on the DT_3 and DT_5 of the results and the STFs calculated thereof is performed to get a more general view. The papers are divided in groups depending on their publisher and open access status. A comparison between open access and non-open access literature of a publisher provides further insights into differences in observable tracking behaviour between the aforementioned groups. A result table and STF, which represent the intersection of all detected hosts in each paper, are generated from the groups.

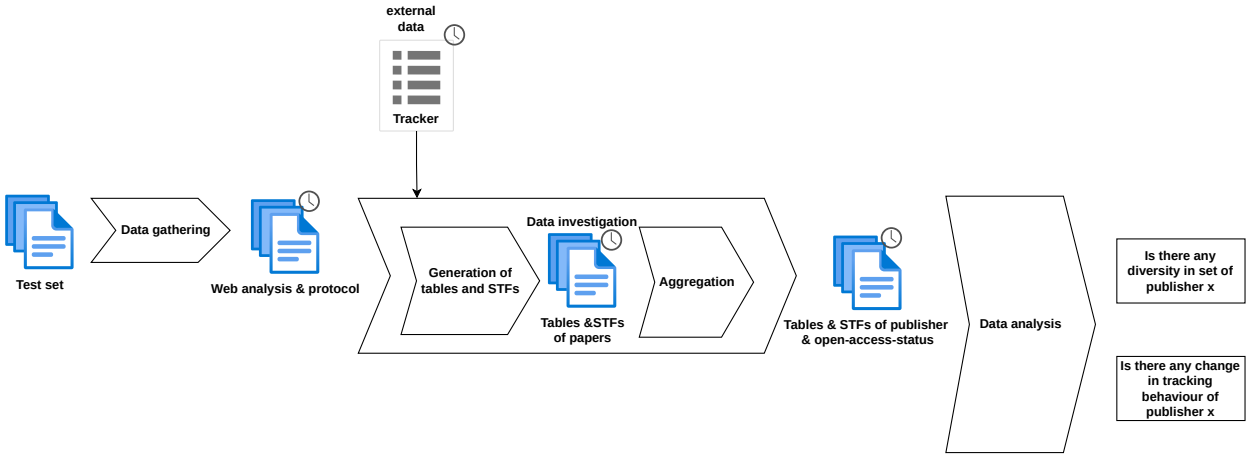


Figure 2. Visualization of the complete conceptual approach.

The Listing 5 in Section VI-A visualizes the aggregation as a pseudo-code algorithm.

C. Data analysis

With the generated result tables and STFs based on the DT_3 and DT_5 data from the investigation step, further examination is performed during the data analysis step, which also entails a detailed evaluation. The trackers detected in the emerging groups of papers are checked for intra-publisher diversity (a deviation from the intersected STF of the publisher; not to be confused with the statistical deviation) within the set of detected trackers (see also Listing 8 in Section V-B3).
Additionally to the intra-publisher analysis, results from different test series are compared by checking the results of the same paper for differences between the test series.
Furthermore, groups of the same publisher but with different open access status are checked for a difference in the set of detected trackers. Last but not least, a comparison between STFs of different versions of the tracker lists is executed for each paper.

Table I from the companion conference article [1] shows an exemplary result table containing the STFs of the publisher ACM and is used to outline the procedure. (Note that in this article, we are focusing exclusively on the web-based retrieval of papers and thus only on web-based Science-Tracking.)

	A-Record 1st party	CNAME 1st Party	A-Record 3rd Party	CNAME 3rd Party
Web	0	0	$3_{PL,N,DT5;PL,N,DT3,L}$	0
	0	0	$2_{PL,N,DT5;PL,N,DT3,L}$	$2_{PL,N,DT5;PL,N,DT3,L}$
	0	0	$6_{PL,N,DT5;UNC,N,DT3,L}$	0
	0	0	$2_{PL,N,DT5;UNC,N,DT3,L}$	$2_{PL,N,DT5;UNC,N,DT3,L}$
	0	0	$1_{UNC,N,DT5;UNC,N,DT3,L}$	$1_{UNC,N,DT5;UNC,N,DT3,L}$
App	0	0	$1_{UNC,T,DT5;PL,T,DT3,L;\ UNC,N,\ DT5;\ PL,\ S,\ DT3,\ L}$	$1_{UNC,T,DT5;PL,T,DT3,L;\ UNC,N,\ DT5;\ PL,\ S,\ DT3,\ L}$
	0	0	$1_{UNC,T,DT5;PL,T,DT3,L;\ UNC,N,\ DT5;\ PL,\ S,\ DT3,\ L}$	0
Email	0	0	$1_{PL,T,DT5;PL,T,DT3,M,\ 1PL,N,DT5;PL,N,DT3,M}$	0

TABLE I. Exemplary Science-Tracking Fingerprint (STF) from the companion conference article [1] of the ACM publisher using the structured semantic description and the syntactical vector formed by element-value pairs.

As a means of evaluating the difference between two STFs, we introduce the STF-deviation as a metric, forming ***Extension E2*** in Section I.
The STF-deviation serves as an estimate to the degree of difference between two STFs. The intention is to generate a value that can be compared to a percentage difference where 0.0 means no difference and 1.0 and above means total difference in tracking behaviour. The STFs in question can be either derived from a publication or an intersection of a group of papers from a publisher. This enables a comparison of paper websites to the collected information of a group. For the following equations we will use $a_{i,j}$ as the value of the cell of a STF A and $b_{i,j}$ is the value of the cell of a reference STF B. The STF-deviation is formed by row-wise comparison of the STFs and summing the relative differences with respect to the size of the respective row difference and the total size of STF B. The latter results in the STF-deviation to be a weighted sum due to the ratio, which is intended to put the row difference in perspective to the total size of STF B.

$$\Delta_{row}(i) = \sum_{j \in StfCol} |a_{i,j} - b_{i,j}|, i \in StfRow \quad (1)$$

$$rowDev(i) = \begin{cases} NaN & if \sum_j b_{i,j} = 0 \\ \frac{\Delta_{row}(i)}{\sum_{j \in StfCol} b_{i,j}} & otherwise \end{cases} \quad (2)$$

$$Dev = \sum_{i \in StfRow} rowDev(i) \cdot \frac{\Delta_{row}(i)}{\sum_{j \in StfRow, k \in StfCol} b_{j,k}} \quad (3)$$

The indices i and j correspond to the cell within the STF without taking the title column and row into account (e.g., i=1 and j=3 corresponds to “first row, A-Record third Party”, pointing to “$3_{PL,N,DT_5;PL,N,DT_3,L}$”). Equation (1) mirrors the total size of the mismatch between the STF’s and references’ row. The total size of the mismatch is put into perspective to the row size of the reference in Equation (2) as a deviation to the row of the references. The summands of the deviation

are weighted to put the deviation of a row into perspective to the total size of the reference STF in Equation (3). This is done with the intention of reducing the distortion due to different sizes of the rows. As one can see, Equation (2) is only partially defined. We decided that in this research only rows from the referenced STF will be taken into account for the deviation to avoid the distortion of the resulting deviation value in Equation (3). That means, that the value of the deviation does not encompass the total deviation but is a measure for the minimum deviation of a STF from another. Furthermore, the STF-deviation is not satiated at 1.0 since, depending on the STFs chosen for comparison, a higher value than 1.0 may be achieved. Whether this issue might be fixable by inversion of the value or is a general problem of the metric, is not clear at this point. Future work should address the issues and aim for a total STF-deviation metric with a more percentage-wise approach. The implementation of the STF-deviation metric is shown in Listing 7 in Section V-B3.

While the calculation of STF-deviation and the collection of comparison results is carried out automated, the results are evaluated as interpretable trends. In the evaluation, the STF-deviation values are interpreted. The interpretations are based on the values themselves and the comparison of values between different analysis groups (intra-publisher, inter-publisher, etc.). There are two general rules for the interpretation:

- Smaller values are interpreted as small deviation, which indicates similar tracking behaviour and greater values vice versa,
- Values similar to a certain analysis group are interpreted as such.

For example, if the values in the intra-publisher comparison group are between values x and y and a STF-deviation of a comparison lies within the interval [x,y], the value is interpreted as being a trend to similar tracking behaviour.
This method of determining whether a STF is similar to the tracking behaviour has further drawbacks:

- interpreting lower values, even zeros, as similar might result in more false positives,
- interpreting higher values as not similar might result in more false negatives.

D. System landscape analysis for Science-Tracking Fingerprint examination

Extending and focusing our research from [1], we eliminate the off-premises examination by hosting our own Webbkoll server inside the examiner's System E1 and thus on-premises. Further, we limit ourselves to web-based access to scientific articles, enabling an in-depth analysis with substantially more tests. Figure 3 shows the altered setup.
It shows both the data flows from the user's perspective and the data flows from a digital forensics perspective. The data flow from the user's perspective consists of using a browser on a computer system that is part of the university's WLAN. In Figure 3 the user activity can be abstracted by the browsers provided by the VM of the examiner's VM DG1. Its network infrastructure can access the OPAC Gateway G1, which then uses the Internet connection of the university to access the publisher's web server delivering the papers (and potentially accessing first and third party trackers).
From the digital forensics perspective the data flow starts by capturing the data traffic at the bridged network interface as DS_N from the examiner's VM DG1. The captured network packets when using the tools Section III-B TShark and the results of using Webbkoll, Website Evidence Collector, Ungoogled Chromium and the script gather_data.py are stored onto mass storage as DS_T (see also Section IV-A). The data from the data gathering step is then transferred to the mass storage DS_T of the analysis workstation AW1 for further investigation and analysis (see also Sections IV-B and IV-C).

Compared to to the system landscape description from the companion article [1], the landscape is also altered by using the Online Public Access Catalogue (OPAC) gateway OPAC G1 hosted by the library system of our university, which routes any searches using the OPAC and provides access to articles under the subscription scheme of our universities' library and allows to answer ***Research Question RQ1*** from Section I. This shows a slightly different flow of data and information but does not prevent Science-Tracking (see Section V). Extending the system landscape with the OPAC gateway enables simulating a typical scientific literature research scenario, which addresses ***Research Question RQ3*** (see Section I) by means of the ***Extension E4*** see (Section I).

V. Implementation of the Automation

This section describes the implemented environment of our research, our analysis tools and components of our automation, the latter forming our ***Extension E5*** (see Section I).

A. System and tool chain

For our research, multiple platforms are used (see Section III-B). The acquisition of research data is performed on the "tester stick" already used in [1], which is in essence a Debian-64-bit-based VM running inside VirtualBox [29] and configured to use a bridged network adapter configured for low noise acquisition of the incoming web traffic, i.e., the system itself and the browser are configured to not actively connect to the network outside the research context; automatic system and browser updates, safebrowsing, certificate updates etc., is disabled.
To show the independence from a particular OS after the data gathering step, the acquired data is processed on Windows10-based PC with an Intel i5-8600k CPU, 16 GB RAM. For both the headless browser and the interactively used browser we employ Ungoogled Chromium [30].
To keep the automation mostly OS-agnostic, the tool chain was implemented in Python 3.12.2, though some adjustments have to be done for the acquisition. This is necessary since the terminating of an asynchronous process needs different signals to be sent, depending on the OS.

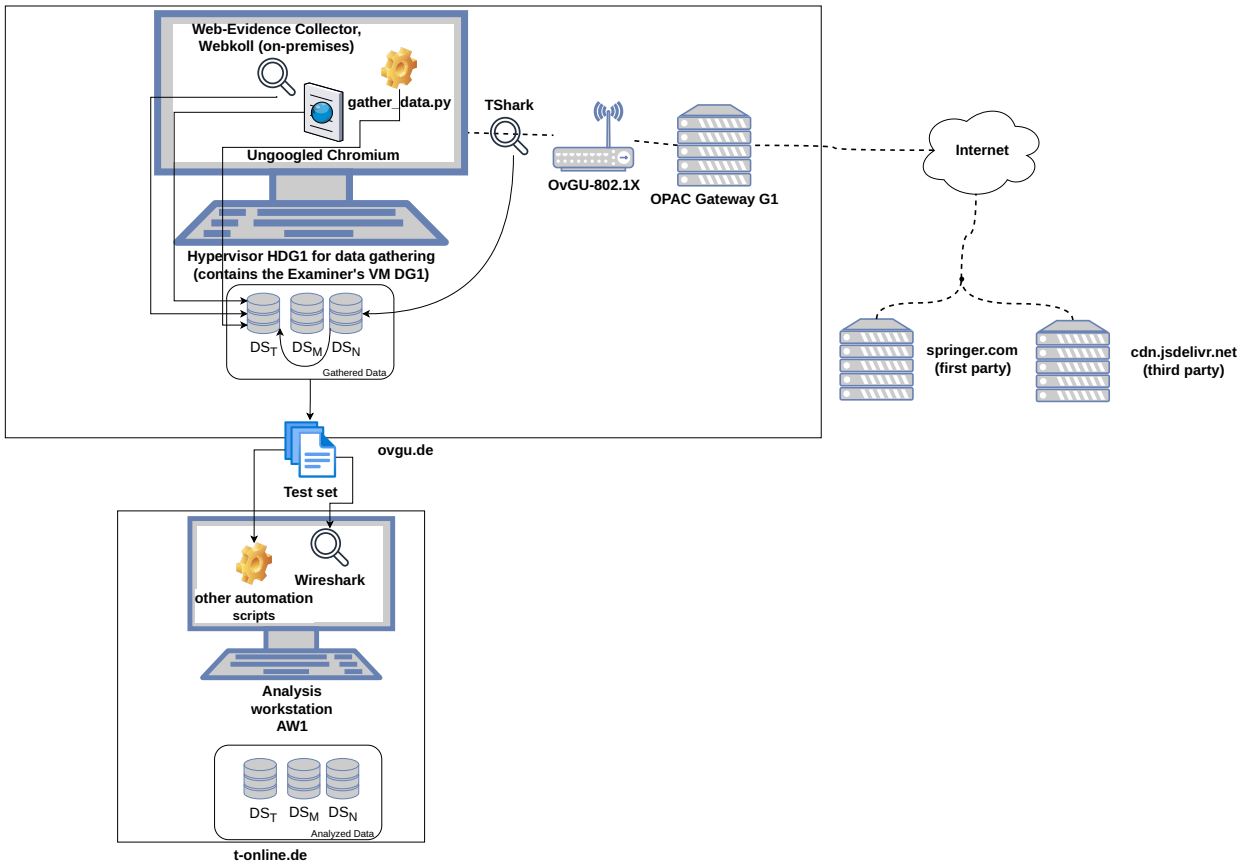


Figure 3. Simplified system landscape analysis for STF examinations visualizing components connections and data flows during the forensic examination, extending and focusing the research from [1]), the dashed lines represent the functional data flow from the user's perspective whilst solid lines represent the data flows of the examination from the digital forensics perspective.

B. Implementation of data acquisition and generation of results

This subsection describes the implementation of the aforementioned investigation concept in Section IV using 10 scripts (6 software tools and 4 automation scripts) in context to their respective steps.

The Webbkoll backend [28] (see Section III-B), is forked for necessary adjustments necessary for the automation, since consistent updating of the user-agent was necessary due to possible detection mechanisms of the publisher websites. The resulting .json file from the analysis is used for further estimation of third party trackers.

The Web-Evidence-Collector [21] (see Section III-B), is adjusted as well for the purposes of automation and a fork was created. Similar to Webbkoll backend, the user-agent is updated but also the tracker lists. A complete overview of the adjustments on both tools can be viewed in the commit list on the respective repositories.

In the following for the sake of brevity, we present pseudocode that represents the actions of our separate python scripts available from [31].

1) Data Gathering: For the implementation of the data gathering, a semi-parallel approach is used. A csv-table containing:

- the URL,
- the OPAC-URL,
- the publisher,
- an alternative URL from a different university,
- the state (open-access, non-open-access),

is provided as input, containing publications with their respective OPAC permalink and publisher website to be called. While the Webbkoll backend and TShark (see Section III-B) are started as an asynchronous process, the calls of the other analysis tools are initiated synchronously.

The TShark process is closed once the other tools completed analysis and restarted once the environment is ready for the next paper.

The Webbkoll backend only has to be terminated after the test series is completed, since the call to initiate the analysis is done by using curl [32].

The paper websites are called randomly with timeouts (randomized, 4-22 seconds) in between, to neither overload the publisher's server nor raise any suspicion, which could interfere with the acquisition process. Additionally, some timeouts are added, since the startup time of the asynchronous tools had to be considered.

The script *gather_data.py* implements the described approach.

```
1 TSharkInterface, PaperWebsiteList

3 startWebkollBackend()
4 randomize(PaperWebsiteList)
5 for paperWebsite in PaperWebsiteList:
6     datetime = now()
7     startTShark(TSharkInterface)
8     webkollOutput <- webkollScan(paperWebsite)
9     TSharkLogOutput = stopTShark()
10    webEvidenceOutput <- webEvidenceCollect(
          paperWebsite)
11    saveOutputsToFilesystem(webkollOutput,
          TSharkLogOutput, webEvidenceOutput)
12 stopWebkollBackend()
```

Listing 2. Pseudo algorithm for gathering data.

2) Data Investigation: The implementation of the generation of data investigation follows the pseudo-code algorithm in Listings 3 to 5. The general structure of the data investigation starts with importing and extracting DT_5 from the gathered sources (see Listing 3, lines 3 to 19). Once all information is gathered, the corresponding process to determine DT_5 and DT_3 matches, including the detection via tracking lists, is performed (see Listing 3, lines 20 to 27). The resulting STF is calculated iteratively for every host. The aggregation follows an iterative approach as well by first forming an intersected result table and generating from this the STF (see Listing 5).

Additional to the modules included in the distribution, the module Scapy [33], version 2.5.0, is used for processing the pcapng files from TShark and getting the required information from the DNS responses. As for its capabilities used in our research, Scapy offers extracting information from a byte stream, e.g., pcapng files, and presenting it in a human-readable format, like Wireshark does in its GUI. The generation utilizes an object-oriented approach to caching the output data from the tools and makes the script more readable.

Listing 3 describes the Generation of the result table and STF. It is implemented as *evaluator.py*.

```
1 Results, STF, Row, Trackerlists

3 WebkollData = parse(WebkollFile)
4 WebEvidenceData = parse(WebEvidenceFile)
5 TSharkData = parse(TSharkLog)

7 Hosts = getAllHosts(WebkollData, WebEvidenceData,
      TSharkData)

9 For every Host in Hosts:
10   If Host in WebkollData then
11     Row <- (WebkollHost: Hostname from WebkollData)
12     Row <- (WebkollIp: Ip from WebkollData)
13   If Host in WebEvidenceData then
14     Row <- (WebEvidenceHost: Hostname from
           WebEvidenceData)
15     Row <- (WebEvidenceParty: Party from
           WebEvidenceData)
16   If Host in TSharkData then
17     Row <- (TSharkHost: Hostname from TSharkData)
18     Row <- (TSharkIp: Ip from TSharkData)
19     Row <- (TSharkType: Type from TSharkData)
20   DT_5 = checkHosts(Row)
21   Row <- DT5
22   DT3 = checkTracker(Row, Trackerlists)
23   Row <- DT3
24   Results <- Row
25   If DT3 atleast UNC then
26     STF = updateSTF(STF, Row)
27     clear(Row)
28 createTable Results, STF
```

Listing 3. Generation of result table and STF.

Listing 4 outlines the update procedure for the generated STFs and is implemented in *evaluator.py* and *aggregation.py*.

```
1 Row, STF

3 If Row is CNAME then
4     STF(Row->DT5,Row->DT3,Row->WebEvidenceParty,A-
          Record) += 1
5     STF(Row->DT5,Row->DT3,Row->WebEvidenceParty,
          CNAME) += 1
6 Else
7     STF(Row->DT5,Row->DT3,Row->WebEvidenceParty,A-
          Record) += 1
8 return STF
```

Listing 4. Updating the STF.

The aggregation of results for multiple papers is shown in Listing 5. It is implemented in the script *aggregation.py*. The scripts *auto_evaluation.py* and *automate_generate_eval_stuff.bat* automate the process of result table and STF generation.

```
1 Results, STF, Row, Trackerlists

3 Papers = getPaperResults(Filter...)
4 For every Paper in Papers:
5   If Results is empty then
6       Results = Paper
7   Else
8       Results = intersect(Results, Paper)
9   If Results is empty then
10      stop

12 STF = evaluateSTF(Results)
13 createTable(Results, STF)
```

Listing 5. Aggregation of results and STFs.

3) Data Analysis: The implementation of the data analysis focuses on the comparison of STFs and estimation of their deviation from each other.

The *fnc_module.py* provides the functions *scan_stf*, reading a STF that was saved to the disk, and *analyze_stfs*, comparing two STFs and generating a report as well as an estimation value for the deviation. The function *analyze_stfs* implements the metric from Section IV-C for grading the deviation of two STFs.

The script *diversity_analysis.py* performs the intra-publisher analysis and is automated over all test sets and publishers with the script *call_diversity_analysis.py*. The two aforementioned scripts focus on the implementation of a intra-publisher analysis.

The remainder of the functionality needed for our research

is implemented in the scripts *inter_pub_diversity_analysis.py*. Those compare two STFs and generating reports. Further, *inter_pub_diversity_auto.py*, automate the process and generating a complete report. That functionality encompasses:

- an inter-publisher analysis within one test series,
- an inter-test-series analysis and an analysis of differences between open access and non open access literature.

All reports are generated as a CSV file and our results can be found in our provided repository at [31].

```
Path, Filters

STF = initializeSTF()
files = listFilesIn(Path)
for file in files:
    if not isDir(file) and filenameStartsWith(file,
        "stf") and satisfiesFilters(file, Filters)
        :
        for row in file:
            category <- determineCategory(row)
            STF <- readRow(category, row)
return STF
```

Listing 6. Pseudo-code algortihm of scan_stf.

```
STF, ReferenceSTF, Title

deviation, totalDeviation, referenceSTFtotalSize =
    0
deviationList = []

stfDifferences = initializeSTF()
report = intializeReport(Title)

for row in rows(stfDifferences):
    rowSizeReferenceSTF = sum(values(ReferenceSTF[
        row]))
    referenceSTFtotalSize += rowSizeReferenceSTF
    stfDifferences <- getDifference(STF[row],
        ReferenceSTF[row])
    differenceRowSize = sum(stfDifferences[row])
    if rowSizeReferenceSTF is 0:
        deviation = 0.0
        report <- stfDifferences, 'not gradable'
    else:
        deviation = abs(differenceRowSize /
            rowSizeReferenceSTF)
        report <- stfDifferences, deviation
    deviationList <- (deviation, differenceRowSize)

for (deviation, differenceRowSize) in deviationList
    :
    totalDeviation += deviation * (
        differenceRowSize / referenceSTFtotalSize)

finishUp(report)
return report, totalDeviation
```

Listing 7. Pseudo-code algorithm of analyse_stfs.

```
Input: TestSeries, Publisher, Version, Reference

intraPublisherReport = initializeReport(Publisher,
    Version)
referenceSTF = scanSTF(Reference)

for paper in TestSeries:
    if isPublisher(paper, Publisher) and isVersion(
        paper, Version):
        stf <- scan_stf(paper, Version)
        paperReport, totalDeviation <- analyse_stfs
            (stf, referenceSTF)
        intraPublisherReport <- addToReport(paper,
            totalDeviation, Version)
        saveToFS(paperReport)

finishUp(intraPublisherReport)
saveToFS(intraPublisherReport)
```

Listing 8. Intra-publisher analysis for a specific test series.

```
FstPaper, SndPaper, FstPublisher, SndPublisher,
    TestSeries, Versions, CompleteReport

for version in Versions:
    fstStf <- scan_stf(getFile(FstPaper, TestSeries
        [0]))
    if SndPaper not undefined:
        sndStf <- scan_stf(getFile(SndPaper,
            TestSeries[0]))
    else:
        sndStf <- scan_stf(getFile(FstPaper,
            TestSeries[1]))
    comparisonReport, deviation <- analyse_stfs(
        fstStf, sndStf)
    saveToFS(comparisonReport)
    if SndPaper not undefined:
        CompleteReport <- addToReport(FstPublisher,
            SndPublisher, version, deviation)
    else:
        CompleteReport <- addToReport(FstPublisher,
            TestSeries, version, deviation)
```

Listing 9. General algortihm for comparing two STFs.

In the following section we evaluate the approach from Section IV in its implementation as described in Section V.

VI. Evaluation

The automation is tested successfully and enables to process a larger amount of literature in a smaller time frame than in [1]. The tool chain enables an almost OS-agnostic automation approach for the generation of the STF (excluding the data gathering step). About 2.1 GB of research data is collected due to the use of the tool chain.

Due to the greater set of publications, some additional insights are gained into the capabilities of the STF. With the automation, a more fine granular examination on the publisher can be performed.

The investigation centres around gaining a greater insight into the possible tracking behaviour on the website of publisher with respect to different points of acquiring a publication and open access status (see Sections IV and V).

The results in Section VI-A to Section VI-G show, that using the OPAC gateway from our university does not prevent tracking, answering ***Research Question RQ1*** (see Section I). Furthermore, the influence of time (t_i, t_{i+1} see [1]) regarding the specific date the tracker lists are acquired, on the recognition/classification of third party hosts is investigated. Also, as an additional comparison to the publishers investigated in [1], the publisher Wiley is also partially added; only in test series 2 and 3 due to mid-experiment inclusion after examination of test series 1.

A. Influence of browser type on tracking

This subsection describes our findings of our research about the influence of the browser type used on tracking behaviour mentioned in Section IV-A and enhances our findings described in the companion article [1] as ***Extension E8*** (see Section I). The recordings of our comparative research are saved in a repository and can be provided on request.

Our first observation shows a difference in the recorded network traffic in all test cases. As for our second observation, there is no clear trend in behaviour depending on the type of browser. In both cases, one type gathered more data than the other.

In the following, we show an example using our list of literature represented in the file paper.csv, which can be viewed in our repository [31]. On paper No.7 the recordings of the graphical browser show, that 5 URLs have been additionally called in comparison to the headless browser. But on paper No.15 only 1 host has been called on the graphical browser. Further information is contained in Figure 4. It can be surmised based on the results of this comparative research that there is an influence of the browser type. We argue for the usage of the headless browser on the grounds of getting results on a larger scale although we are potentially missing some trackers by using the headless browser.

Index	# Tracker (Headless Browser)	# Tracker (GUI Browser)	Difference	Relative difference Headless : GUI
0	2	1	-1	200.00%
7	2	7	5	28.57%
15	2	1	-1	200.00%
25	6	6	0	100.00%
33	7	12	5	58.33%
41	12	11	-1	109.09%
49	13	12	-1	108.33%
63	18	15	-3	120.00%
69	6	2	-4	300.00%

Figure 4. Results of the probe for tracking based on browser type.

Finding the source of the different tracking behaviour, however, is a very valid research goal for future work.

B. Time dependency of tracker lists

This research enhances the companion article [1] as ***Extension E6*** (see)Section I). During our investigation, tracker lists are downloaded from every provider at the last time of change before the acquisition.

To check if the classification behaviour changes over a short period of time, the tracker list versions are grouped by a date that signifies the last change on one of the lists before the recording, and applied on every test series.

The generated reports, e.g., Figure 14, show the same results and thus being independent of the version of the tracker lists for the tests conducted. Future research should examine the time dependency over a broader time span.

Due to these results, some of the result tables will be abridged due to there not being any benefit for showing the results with respect to every version of the tracker lists. The full set of results can be found at [31].

C. Intra-publisher diversity

By comparing the result tables and STF of single papers with the intersected results of a test series during our research enhancing our companion article [1] with the ***Extension E3*** (see Section I), a diversity within the third party hosts classified as probable tracker (classified via external data in the form of tracker lists, see Section III-B) is observed in all but one publisher, namely Springer. Figures 5 to 11 show exemplary, how strongly the STF of a paper can differ from the intersected STF of its publisher in comparison to its peers.

The papers from the publisher ACM show the second strongest intra-publisher diversity. Throughout every test series, there is no paper that does not match the intersection completely. Also, in comparison to Figure 12, every deviation value is higher. An interesting detail is that some deviation values are the same and on closer inspection with the STFs, the STFs are the same. While this is no proof that the same deviation signals an equal STF, it may indicate heterogeneity in the set of STFs.

The following Figure 5 shows the intra publisher diversity based on the STF-deviation for the publisher ACM according to the test series conducted at 20/02/2024.

Report for publisher ACM from test series 20240220 with tracking list version -20240225	
Entry	STF-deviation to publisher STF
33-20240220T132709-ACM-NonOpenAccess	0.866666666666667
34-20240220T132802-ACM-NonOpenAccess	2.016666666666667
35-20240220T132853-ACM-NonOpenAccess	0.683333333333333
36-20240220T132938-ACM-NonOpenAccess	0.816666666666667
37-20240220T133007-ACM-NonOpenAccess	0.816666666666667
38-20240220T133036-ACM-NonOpenAccess	0.816666666666667
39-20240220T133104-ACM-NonOpenAccess	0.816666666666667
40-20240220T133133-ACM-NonOpenAccess	0.816666666666667
STF-deviation may not encompass the complete deviation due to constraints	

Figure 5. Intra-publisher comparison results for ACM (test series 2024-02-20).

The following Figure 6 shows the intra publisher diversity based on the STF-deviation for the publisher ACM according to the test series conducted at 12/03/2024.

Report for publisher ACM from test series 20240312 with tracking list version -20240220	
Entry	STF-deviation to publisher STF
33-20240312T111922-ACM-NonOpenAccess	1.01785714285714
34-20240312T105414-ACM-NonOpenAccess	1.85714285714286
35-20240312T111046-ACM-NonOpenAccess	0.875
36-20240312T105540-ACM-NonOpenAccess	0.875
37-20240312T104700-ACM-NonOpenAccess	4.16071428571429
38-20240312T110933-ACM-NonOpenAccess	0.732142857142857
39-20240312T103100-ACM-NonOpenAccess	1.85714285714286
40-20240312T103318-ACM-NonOpenAccess	0.732142857142857
63-20240312T111455-ACM-NonOpenAccess	0.571428571428571
64-20240312T111550-ACM-NonOpenAccess	0.714285714285714
65-20240312T110535-ACM-NonOpenAccess	0.571428571428571
66-20240312T105054-ACM-NonOpenAccess	0.589285714285714
67-20240312T103224-ACM-NonOpenAccess	0.571428571428571
68-20240312T102238-ACM-NonOpenAccess	0.571428571428571
STF-deviation may not encompass the complete deviation due to constraints	

Figure 6. Intra-publisher comparison results for ACM (test series 2024-03-12).

The following Figure 7 shows the intra publisher diversity based on the STF-deviation for the publisher ACM according to the test series conducted at 25/03/2024.

From Figures 8 and 9 it can be assumed, that in that specific test series the STFs of the papers from Elsevier show a strong difference in observed tracking behaviour and a mentionable

intersection between the STFs could not be formed.

Report for publisher ACM from test series 20240325 with tracking list version -20240220	
Entry	STF-deviation to publisher STF
33-20240325T090906-ACM-NonOpenAccess	0.746666666666667
34-20240325T090527-ACM-NonOpenAccess	1.32
35-20240325T095312-ACM-NonOpenAccess	0.586666666666667
36-20240325T092837-ACM-NonOpenAccess	0.633333333333333
37-20240325T093827-ACM-NonOpenAccess	0.72
38-20240325T092057-ACM-NonOpenAccess	0.586666666666667
39-20240325T091234-ACM-NonOpenAccess	0.586666666666667
40-20240325T094511-ACM-NonOpenAccess	0.986666666666667
63-20240325T091321-ACM-NonOpenAccess	0.446666666666667
64-20240325T093740-ACM-NonOpenAccess	0.68
65-20240325T091413-ACM-NonOpenAccess	0.68
66-20240325T093600-ACM-NonOpenAccess	0.313333333333333
67-20240325T094732-ACM-NonOpenAccess	0.533333333333333
68-20240325T092725-ACM-NonOpenAccess	0.466666666666667
STF-deviation may not encompass the complete deviation due to constraints	

Figure 7. Intra-publisher comparison results for ACM (test series 2024-03-25).

This might also be connected to the findings in Section VI-E, as they show a difference in tracking behaviour between open access and non-open access groups.
It should also be mentioned that there are STFs of Elsevier publications in test series 2024-02-20 but since no tracking data is gathered for some publications, the intersected STF was empty. Therefore, an analysis on the intra-publisher diversity is impossible for this test series. The following Figure 8 shows the intra publisher diversity based on the STF-deviation for the publisher Elsevier according to the test series conducted at 12/03/2024.

Report for publisher Elsevier from test series 20240312 with tracking list version -20240225	
Entry	STF-deviation to publisher STF
07-20240312T110809-Elsevier-OpenAccess	0
08-20240312T112054-Elsevier-OpenAccess	0
09-20240312T102340-Elsevier-OpenAccess	0
10-20240312T105904-Elsevier-OpenAccess	0
11-20240312T111856-Elsevier-OpenAccess	0
12-20240312T111348-Elsevier-OpenAccess	0
13-20240312T103144-Elsevier-OpenAccess	0
14-20240312T105500-Elsevier-OpenAccess	0
41-20240312T110848-Elsevier-NonOpenAccess	36
42-20240312T103819-Elsevier-NonOpenAccess	36
43-20240312T103017-Elsevier-NonOpenAccess	36
44-20240312T104816-Elsevier-NonOpenAccess	36
45-20240312T102039-Elsevier-NonOpenAccess	36
46-20240312T111303-Elsevier-NonOpenAccess	36
47-20240312T110328-Elsevier-NonOpenAccess	16
48-20240312T102738-Elsevier-NonOpenAccess	36
STF-deviation may not encompass the complete deviation due to constraints	

Figure 8. Intra-publisher comparison results for Elsevier (test series 2024-03-12).

The following Figure 9 shows the intra publisher diversity based on the STF-deviation for the publisher Elsevier according to the test series conducted at 25/03/2024.

Report for publisher Elsevier from test series 20240325 with tracking list version -20240225	
Entry	STF-deviation to publisher STF
07-20240325T092023-Elsevier-OpenAccess	0
08-20240325T094826-Elsevier-OpenAccess	0
09-20240325T090454-Elsevier-OpenAccess	0
10-20240325T095023-Elsevier-OpenAccess	0
11-20240325T093314-Elsevier-OpenAccess	0
12-20240325T100533-Elsevier-OpenAccess	0
13-20240325T100653-Elsevier-OpenAccess	0
14-20240325T100721-Elsevier-OpenAccess	0
41-20240325T095704-Elsevier-NonOpenAccess	25
42-20240325T092158-Elsevier-NonOpenAccess	25
43-20240325T090017-Elsevier-NonOpenAccess	25
44-20240325T090137-Elsevier-NonOpenAccess	25
45-20240325T091747-Elsevier-NonOpenAccess	25
46-20240325T091202-Elsevier-NonOpenAccess	25
47-20240325T095848-Elsevier-NonOpenAccess	9
48-20240325T094142-Elsevier-NonOpenAccess	25
STF-deviation may not encompass the complete deviation due to constraints	

Figure 9. Intra-publisher comparison results for Elsevier (test series 2024-03-25).

The findings for the paper from the publisher IEEE show almost no intra-publisher diversity throughout the whole test series. Still, even in the case shown in Figure 12 the intra-publisher diversity is low in comparison to other publishers like ACM or Elsevier. This might indicate that the intersected STF of IEEE encompasses almost every detected host of the test series or in the case Figures 10 and 11 every host. The following Figure 10 shows the intra publisher diversity based on the STF-deviation for the publisher IEEE according to the test series conducted at 20/02/2024.

Report for publisher IEEE from test series 20240325 with tracking list version -20240225	
Entry	STF-deviation to publisher STF
25-20240325T090608-IEEE-NonOpenAccess	0.17948717948718
26-20240325T093115-IEEE-NonOpenAccess	0.138461538461538
27-20240325T100141-IEEE-NonOpenAccess	0.17948717948718
28-20240325T092235-IEEE-NonOpenAccess	0.153846153846154
29-20240325T095450-IEEE-NonOpenAccess	0.17948717948718
30-20240325T094009-IEEE-NonOpenAccess	0.17948717948718
31-20240325T094856-IEEE-NonOpenAccess	0.17948717948718
32-20240325T091507-IEEE-NonOpenAccess	0.153846153846154
69-20240325T093348-IEEE-NonOpenAccess	0.17948717948718
70-20240325T094615-IEEE-NonOpenAccess	0.17948717948718
71-20240325T100317-IEEE-NonOpenAccess	0.00512820512820513
72-20240325T091819-IEEE-NonOpenAccess	0.153846153846154
73-20240325T090318-IEEE-NonOpenAccess	0.17948717948718
74-20240325T092542-IEEE-NonOpenAccess	0.153846153846154
STF-deviation may not encompass the complete deviation due to constraints	

Figure 10. Intra-publisher comparison results for IEEE (test series 2024-02-20).

The following Figure 11 shows the intra publisher diversity based on the STF-deviation for the publisher IEEE according to the test series conducted at 12/03/2024.

Report for publisher IEEE from test series 20240220 with tracking list version -20240313	
Entry	STF-deviation to publisher STF
25-20240220T132145-IEEE-NonOpenAccess	0
26-20240220T132227-IEEE-NonOpenAccess	0
27-20240220T132307-IEEE-NonOpenAccess	0
28-20240220T132350-IEEE-NonOpenAccess	0
29-20240220T132431-IEEE-NonOpenAccess	0
30-20240220T132509-IEEE-NonOpenAccess	0
31-20240220T132546-IEEE-NonOpenAccess	0
32-20240220T132630-IEEE-NonOpenAccess	0
STF-deviation may not encompass the complete deviation due to constraints	

Figure 11. Intra-publisher comparison results for IEEE (test series 2024-03-12).

The following Figure 12 shows the intra publisher diversity based on the STF-deviation for the publisher IEEE according to the test series conducted at 25/03/2024.

Report for publisher IEEE from test series 20240312 with tracking list version -20240225	
Entry	STF-deviation to publisher STF
25-20240312T105202-IEEE-NonOpenAccess	0
26-20240312T103415-IEEE-NonOpenAccess	0
27-20240312T104203-IEEE-NonOpenAccess	0
28-20240312T103535-IEEE-NonOpenAccess	0
29-20240312T102921-IEEE-NonOpenAccess	0
30-20240312T110040-IEEE-NonOpenAccess	0
31-20240312T104502-IEEE-NonOpenAccess	0
32-20240312T111726-IEEE-NonOpenAccess	0
69-20240312T110709-IEEE-NonOpenAccess	0
70-20240312T101858-IEEE-NonOpenAccess	0
71-20240312T105943-IEEE-NonOpenAccess	0
72-20240312T104558-IEEE-NonOpenAccess	0
73-20240312T111207-IEEE-NonOpenAccess	0
74-20240312T104102-IEEE-NonOpenAccess	0
STF-deviation may not encompass the complete deviation due to constraints	

Figure 12. Intra-publisher comparison results for IEEE (test series 2024-03-25).

In the set of publications of the publisher Springer during the test period no diversity in the set of classified trackers is observable, see Figure 13. This behaviour within STFs

of publications from Springer is observable throughout the complete test series and during our tests is unique to the publisher Springer.

Report for publisher Springer from test series 20240325 with tracking list version -20240220	
Entry	STF-deviation to publisher STF
00-20240325T094427-Springer-OpenAccess	0
01-20240325T100448-Springer-OpenAccess	0
02-20240325T093657-Springer-OpenAccess	0
03-20240325T091123-Springer-OpenAccess	0
04-20240325T095613-Springer-OpenAccess	0
05-20240325T090056-Springer-OpenAccess	0
06-20240325T091049-Springer-OpenAccess	0
15-20240325T095217-Springer-NonOpenAccess	0
16-20240325T091009-Springer-NonOpenAccess	0
17-20240325T093925-Springer-NonOpenAccess	0
18-20240325T100615-Springer-NonOpenAccess	0
19-20240325T100938-Springer-NonOpenAccess	0
20-20240325T095418-Springer-NonOpenAccess	0
21-20240325T091949-Springer-NonOpenAccess	0
22-20240325T100748-Springer-NonOpenAccess	0
23-20240325T090719-Springer-NonOpenAccess	0
24-20240325T101127-Springer-NonOpenAccess	0
STF-deviation may not encompass the complete deviation due to constraints	

Figure 13. Intra-publisher comparison results for Springer (test series 2024-03-25).

In the following we compare different publishers using the proposed STF-deviation metric.

D. Inter-test series diversity

Besides checking for diversity within the set of probable tracker within one test series, the results in between test series are compared for any observable difference. This enables the detection of a potential diversity in the time dimension, enhancing our research from the companion article [1] as the ***Extension E3*** (see Section I).
Figure 14 shows the results of the comparison within our tests.

Report of inter-test-series comparison		
Compared test series of publisher	Tracker list version	STF-deviation
Springer-20240220-20240312	20240220	0
Springer-20240220-20240312	20240225	0
Springer-20240220-20240312	20240313	0
Springer-20240220-20240325	20240220	0
Springer-20240220-20240325	20240225	0
Springer-20240220-20240325	20240313	0
IEEE-20240220-20240312	20240220	0
IEEE-20240220-20240312	20240225	0
IEEE-20240220-20240312	20240313	0
IEEE-20240220-20240325	20240220	0.17948717948718
IEEE-20240220-20240325	20240225	0.17948717948718
IEEE-20240220-20240325	20240313	0.17948717948718
ACM-20240220-20240312	20240220	0.142857142857143
ACM-20240220-20240312	20240225	0.142857142857143
ACM-20240220-20240312	20240313	0.142857142857143
ACM-20240220-20240325	20240220	0.146666666666667
ACM-20240220-20240325	20240225	0.146666666666667
ACM-20240220-20240325	20240313	0.146666666666667
Springer-20240312-20240325	20240220	0
Springer-20240312-20240325	20240225	0
Springer-20240312-20240325	20240313	0
Elsevier-20240312-20240325	20240220	0
Elsevier-20240312-20240325	20240225	0
Elsevier-20240312-20240325	20240313	0
IEEE-20240312-20240325	20240220	0.17948717948718
IEEE-20240312-20240325	20240225	0.17948717948718
IEEE-20240312-20240325	20240313	0.17948717948718
ACM-20240312-20240325	20240220	0.0133333333333333
ACM-20240312-20240325	20240225	0.0133333333333333
ACM-20240312-20240325	20240313	0.0133333333333333
Wiley-20240312-20240325	20240220	0
Wiley-20240312-20240325	20240225	0
Wiley-20240312-20240325	20240313	0

Figure 14. Results of the comparison of intersected STFs between test series.

In regard to the inter-test series diversity, it is observed that, except for the publisher Springer, every publisher has between at least two test series differences in the intersected STFs, see Figure 14.
This might indicate that changes in tracking behaviour reflect on the STF and can therefore be noticed by the application of the STF.

E. Intra-publisher differences for open access and non-open access papers (OA/NOA)

The investigation of possible differences within the observed trackers answers the ***Research Question RQ4*** (see Section I). It is, however, limited by the constraints of our approach, environment and tools. For instance, OPAC only listed open access publications from the publishers Springer and Elsevier. IEEE and ACM do feature open access publications, but, at least in the case of IEEE, during our tests open access publications are not offered on the publisher's usual website (e.g., IEEE Xplore) but rather a platform specifically for open access publications. ACM itself offers open access literature through searching specifically for it within OPAC results in matches (e.g., using filters for publisher and keyword *open access* or *non-open access*). This necessitates specialized queries.
In addition, a problem is encountered with Webbkoll and Elsevier open access publications, which results in failure to acquire analysis data, and therefore no tracker could be classified plausible for DT_3 or DT_5. From the available analysis data for Elsevier publications a difference in classified trackers between open access and non-open access publications could be observed, see Figure 15.
As for the paper from the publisher Springer, no deviations were observable in the data sets.

Report of open access to non open access stf		
Publisher	Tracker list version	STF-deviation
Test series-20240220		
Springer-Springer	20240220	0
Springer-Springer	20240225	0
Springer-Springer	20240313	0
Test series-20240312		
Springer-Springer	20240220	0
Springer-Springer	20240225	0
Springer-Springer	20240313	0
Elsevier-Elsevier	20240220	0.53030303030303
Elsevier-Elsevier	20240225	0.53030303030303
Elsevier-Elsevier	20240313	0.53030303030303
Test series-20240325		
Springer-Springer	20240220	0
Springer-Springer	20240225	0
Springer-Springer	20240313	0
Elsevier-Elsevier	20240220	0.533333333333333
Elsevier-Elsevier	20240225	0.533333333333333
Elsevier-Elsevier	20240313	0.533333333333333

Figure 15. Results of the comparison of open access to non-open access literature.

Future work should point to an enhanced environment and tools to address the existing challenges.

F. Inter-publisher difference

With the automated approach, a comparison of the intersected STFs of different publishers is performed successfully, enhancing the findings from our companion article [1] as the ***Extension E1*** (see Section I). Figure 16

shows an abridged version of the complete reports, since there is no need to consider the different versions of tracking lists due to the mentioned points in Section VI-B.

Report of inter-publisher comparison		
Compared publishers	Tracker list version	STF-deviation
Test series-20240220		
Springer-IEEE	20240313	0.888235294117647
Springer-ACM	20240313	0.616666666666667
IEEE-ACM	20240313	1.66666666666667
Test series-20240312		
Springer-Elsevier	20240313	1
Springer-IEEE	20240313	0.888235294117647
Springer-ACM	20240313	0.875
Springer-Wiley	20240313	0.701234567901235
Elsevier-IEEE	20240313	1
Elsevier-ACM	20240313	0.75
Elsevier-Wiley	20240313	0.87037037037037
IEEE-ACM	20240313	1.35714285714286
IEEE-Wiley	20240313	1.26172839506173
ACM-Wiley	20240313	0.530864197530864
Test series-20240325		
Springer-Elsevier	20240313	1
Springer-IEEE	20240313	0.871794871794872
Springer-ACM	20240313	0.88
Springer-Wiley	20240313	0.721739130434783
Elsevier-IEEE	20240313	0.866666666666667
Elsevier-ACM	20240313	0.766666666666667
Elsevier-Wiley	20240313	0.847826086956522
IEEE-ACM	20240313	1.28666666666667
IEEE-Wiley	20240313	0.847826086956522
ACM-Wiley	20240313	0.565217391304348

Figure 16. Results of the inter-publisher comparison (abridged).

A complete version with all tracker list version can be found in [31]. The results of Figure 16 show that there is a noticeable difference in tracking behaviour between publishers, which could give hints/leads towards identifying specific publishers based on their tracking behaviour (see also our companion conference article [1]. It can be assumed, based on our results, that the tracking behaviour may strongly differ from publisher to publisher. Future research on an even larger scale (both in number of papers and the time span observed) is needed to have a qualified opinion as to how discriminating the STF-deviation with respect to publishers is.

The full set of tables is available under [31].

G. Addendum Wiley

The publisher Wiley is additionally investigated to expand our group of subjects using the same setup and procedures, enhancing the findings from our companion article [1] as the ***Extension E7*** (see Section I). As Wiley is added mid-investigation, publications of it are only considered in the second and third test series. The following Figure 17 shows the intra-publisher comparison results for the publisher Wiley from the test series conducted at 12/03/2024.

Report for publisher Wiley from test series 20240312 with tracking list version -20240313	
Entry	STF-deviation to publisher STF
49-20240312T102816-Wiley-NonOpenAccess	0.107407407407407
50-20240312T103708-Wiley-NonOpenAccess	0.619753086419753
51-20240312T105801-Wiley-NonOpenAccess	0.619753086419753
52-20240312T110417-Wiley-NonOpenAccess	0.716049382716049
53-20240312T105301-Wiley-NonOpenAccess	0.619753086419753
54-20240312T102632-Wiley-NonOpenAccess	0.619753086419753
55-20240312T102418-Wiley-NonOpenAccess	0.619753086419753
56-20240312T105624-Wiley-NonOpenAccess	0.619753086419753
57-20240312T104942-Wiley-NonOpenAccess	0.619753086419753
58-20240312T102130-Wiley-NonOpenAccess	0.619753086419753
59-20240312T104258-Wiley-NonOpenAccess	0.619753086419753
60-20240312T103905-Wiley-NonOpenAccess	0.619753086419753
61-20240312T110215-Wiley-NonOpenAccess	0.619753086419753
62-20240312T112207-Wiley-NonOpenAccess	0.619753086419753
STF-deviation may not encompass the complete deviation due to constraints	

Figure 17. Intra-publisher comparison results for Wiley (test series 2024-03-12).

The following Figure 18 shows the intra-publisher comparison results for the publisher Wiley from the test series conducted at 25/03/2024.

Report for publisher Wiley from test series 20240325 with tracking list version -20240313	
Entry	STF-deviation to publisher STF
49-20240325T091647-Wiley-NonOpenAccess	0.71304347826087
50-20240325T093450-Wiley-NonOpenAccess	0.71304347826087
51-20240325T094327-Wiley-NonOpenAccess	0.126086956521739
52-20240325T100835-Wiley-NonOpenAccess	0.126086956521739
53-20240325T090801-Wiley-NonOpenAccess	0.71304347826087
54-20240325T095933-Wiley-NonOpenAccess	0.71304347826087
55-20240325T090211-Wiley-NonOpenAccess	0.71304347826087
56-20240325T101015-Wiley-NonOpenAccess	0.71304347826087
57-20240325T095744-Wiley-NonOpenAccess	0.71304347826087
58-20240325T094219-Wiley-NonOpenAccess	0
59-20240325T100033-Wiley-NonOpenAccess	0.71304347826087
60-20240325T092427-Wiley-NonOpenAccess	0.71304347826087
61-20240325T092953-Wiley-NonOpenAccess	0
62-20240325T095107-Wiley-NonOpenAccess	0.71304347826087
STF-deviation may not encompass the complete deviation due to constraints	

Figure 18. Intra-publisher comparison results for Wiley (test series 2024-03-25).

In Figures 17 and 18 it is shown that there is an intra-publisher diversity within the tracking. The deviations seem to form a middle ground between ACM and IEEE, compared to the findings in Section VI-C. Besides being analysed for intra-publisher diversity, an inter-publisher comparison as well as an inter-test series comparison is conducted. Their results are shown in Figures 14 and 16 and indicate, that there was no significant difference in the tracking behaviour over time, but the tracking behaviour deviates from other publishers. Since no open access publications from Wiley in OPAC are to be found by filtering and keyword search during our tests, no examinations with respect to open-access status are conducted at the time of the research.
While those results are not a full addition to the test series, they still show a tendency and underline the unique result position for the publisher Springer so far.

VII. Conclusion and Future Work

In this article we extended the work from the companion conference article [1] centred around the topic of Science-Tracking and the usage of the Science-Tracking Fingerprint (STF) as a means to gain hints for the originator of the tracking. The extension covers 8 separate aspects.
First we altered the system landscape by measuring the amount of Science-Tracking behind our universities' Online Public Access Catalog (OPAC) in order to see whether the Science-Tracking is altered by tunnelling our paper requests and downloads through that system. This is not

the case according to our current results. Even after placing queries through this OPAC system, tracking by the publishers still takes place. We swapped broadness for detail and thus restricted ourselves to the examination of Web-based Science-Tracking.

Secondly, as placed in the future work section of [1], we automated the processes for the detection of Science-Tracking and the calculation of the STF. In total 10 scripts (6 Software tools and 4 automation scripts) that cover mostly the steps of data gathering and data investigation were released as Open Source. We also changed the number of lists of known trackers from originally 1 to 3 to increase the hit ratio for known tracker domains. We were able to examine 60 papers from 4 selected publishers.

Enabled by the automatization and larger numbers of STF-based examinations as a result (60 in total for all examinations), we could observe multiple documents from an individual publisher at 3 different points in time to obtain a measure for the intra-publisher diversity using the Science-Tracking Fingerprint. Our results show for 3 of the 4 publishers there is a notable diversity between the third party hosts suspected to be trackers.

The STF-deviation metric introduced in this paper allows for the comparison of the differences between STFs of different publishers (inter-publisher comparison). The first results show a noticeable difference between the tracking behaviour of the different publishers, giving hope to idea that the publishers could be distinguished from one another and the STF and STF-deviation could give first hints/leads towards identifying a publisher by its tracking behaviour.

We have shown that the tracking behaviour of publishers can differ whether their papers are accessed using an interactive browser as compared to a headless browser. Although these are first results, this points towards interesting research topics to find the cause and mechanisms for detecting the browser type.

We could show that for the duration of our tests the tracking lists used to classify third party hosts as trackers did not change noticeably for the trackers employed by the publishers under examination. We still argue for maintaining the procedure keeping the possibility to check against updated lists of trackers.

Our results highlight the need for future work with regards to the examination environment. First results show there are differences between open access and non-open access papers for some publishers during our tests.

The inclusion of the publisher Wiley, albeit late in the research and lacking open access papers with the universities' OPAC gateway, bolstered our research and showed an intra publisher diversity and inter publisher differences within our tests .

The introduction of the STF-deviation metric allowed for the evaluation of the intra and inter publisher differences.

Future work should address the shortcomings of the STF-deviation metric:

- Not encompassing the total deviation of a STF
- Distortion of the STF-deviation, see Figures 8 and 9
- Limiting the value of the STF-deviation to a range of [0,1], to make it more interpretable
- Reducing the possibility of false positives and negatives

The source of the altered tracking behaviour of interactive vs. headless browsers should be identified and this behaviour mitigated. This would allow for a better quality of the results. The time span for observing changes in tracker lists for relevant tracker entries should be expanded to yield more insights into the relevance of a retrospective evaluation of tracking.

The setup and the software needs adaption to incorporate more sources for the comparison of open access vs. non-open access papers, e.g., the flexibility to add other publishers websites (some publishers have different sites for open and non-open access papers).

Also, some tools (e.g., Webbkoll) are barred from accessing some publisher websites, here mitigation to circumvent the restrictions or alternative tools could be a focus of future research.

ACKNOWLEDGEMENTS

The research from Stefan Kiltz, Robert Altschaffel and Jana Dittmann is partly funded by the EFRE project "CyberSecurity-Verbund LSA II – Prävention, Detektion und Reaktion mit Open Source-Perspektiven" https://forschung-sachsen-anhalt.de/project/cybersecurity-verbund-lsa-praevention-27322.

REFERENCES

[1] S. Kiltz, R. Altschaffel, and J. Dittmann, "Science-tracker fingerprinting with uncertainty: Selected common characteristics of publishers from network to application trackers on the example of web, app and email," in *Proceedings of the Seventeenth International Conference on Emerging Security Information, Systems and Technologies (Securware)*, Porto, Portugal, 2023, pp. 88–97.

[2] Deutsche Forschungsgemeinschaft, "Data tracking in research: aggregation and use or sale of usage data by academic publishers," (last access 2024.11.29). [Online]. Available: https://www.dfg.de/resource/blob/174924/d99b797724796bc1a137fe3d6858f326/datentracking-papier-en-data.pdf

[3] E. Bettinger, M. Bursic, and A. Chandler, "Disrupting the digital status quo: Why and how to staff for privacy in academic libraries," (last access 2024.11.29). [Online]. Available: https://publish.illinois.edu/licensingprivacy/files/2023/06/Whitepaper-on-Privacy-Staffing-Licensing-Privacy.pdf

[4] M. Bambot, "How we hacked the sourcecon 2018 attendee list in 2 hours - by murtaza bambot - medium," (last access 2024.11.29). [Online]. Available: https://medium.com/@MurtazaBambot/how-we-hacked-the-sourcecon-2018-attendee-list-in-2-hours-645bf26d2825

[5] R. Altschaffel, S. Kiltz, T. Lucke, and J. Dittmann, "Introduction to being a privacy detective: Investigating and comparing potential privacy violations in mobile apps using forensic methods," in *Proceedings of the Fourteenth International Conference on Emerging Security Information, Systems and Technologies (Securware)*, Valencia, Spain, 2020, pp. 60–68.

[6] University Library of the Otto von Guericke University Magdeburg, "OPC4 - start/welcome," (last access 2024.11.29). [Online]. Available: https://opac.lbs-magdeburg.gbv.de/DB=1/LNG=EN/

[7] S. Kiltz, "Data-centric examination approach (DCEA) for a qualitative determination of error, loss and uncertainty in digital and digitised forensics," Ph.D. dissertation, Otto-von-Guericke-University, Magdeburg, Germany, 2020, (last access 2024.11.29). [Online]. Available: https://opendata.uni-halle.de/bitstream/1981185920/34842/1/Kiltz_Stefan_Dissertation_2020.pdf

[8] W. Christl, "Corporate surveillance in everyday life," (last access 2024.11.29). [Online]. Available: https://crackedlabs.org/dl/CrackedLabs_Christl_CorporateSurveillance.pdf

[9] H. Mildebrath, "Unpacking 'commercial surveillance': The state of tracking," (last access 2024.11.29). [Online]. Available: https://www.europarl.europa.eu/RegData/etudes/BRIE/2022/739266/EPRS_BRI(2022)739266_EN.pdf

[10] N. Samarasinghe and M. Mannan, "Towards a global perspective on web tracking," *Computers & Security*, vol. 87, p. 101569, 2019, (last access 2024.11.29). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404818314007

[11] K. Sim, H. Heo, and H. Cho, "Combating web tracking: Analyzing web tracking technologies for user privacy," *Future Internet*, vol. 16, no. 10, 2024. [Online]. Available: https://www.mdpi.com/1999-5903/16/10/363

[12] R. Pan and A. Ruiz-Martínez, "Evolution of web tracking protection in chrome," *Journal of Information Security and Applications*, vol. 79, p. 103643, 2023, (last access 2024.11.29). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212623002272

[13] B. Krupp, J. Hadden, and M. Matthews, "An analysis of web tracking domains in mobile applications," in *Proceedings of the 13th ACM Web Science Conference 2021*, ser. WebSci '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 291–298, (last access 2024.11.29). [Online]. Available: https://doi.org/10.1145/3447535.3462507

[14] R. Altschaffel, M. Beurskens, J. Dittmann, W. Horstmann, S. Kiltz, G. Lauer, J. Ludwig, B. Mittermaier, and K. Stump, "Data tracking and DEAL: On the 2022/2023 negotiations and the consequences for academic libraries," *Recht und Zugang (RuZ)*, vol. 5, no. 1, Nov. 2024, (last access 2024.11.29). [Online]. Available: https://doi.org/10.5281/zenodo.14006196

[15] C. Hanson, "User tracking on academic publisher platforms," (last access 2024.11.29). [Online]. Available: https://www.codyh.com/writing/tracking.html

[16] European Commission and Directorate-General for Communications Networks, Content and Technology, C. Armitage, N. Botton, L. Dejeu-Castang, and L. Lemoine, *Study on the impact of recent developments in digital advertising on privacy, publishers and advertisers – Final report*. Publications Office of the European Union, 2023.

[17] Volatility Foundation, "Github - volatilityfoundation/volatility3: Volatility 3.0 development," (last access 2024.11.29). [Online]. Available: https://github.com/volatilityfoundation/volatility3?tab=readme-ov-file

[18] S. Englehardt and A. Narayanan, "Online tracking: A 1-million-site measurement and analysis," in *Proceedings of ACM CCS 2016*, 2016, pp. 1388–1401.

[19] Dataskydd.net Sverige, "Analyze — webbkoll - dataskydd.net," (last access 2024.11.29). [Online]. Available: https://webbkoll.dataskydd.net/en

[20] Wireshark Foundation, "Wireshark - go deep," (last access 2024.11.29). [Online]. Available: https://www.wireshark.org/

[21] European Data Protection Supervisor, "European data protection supervisor / website-evidence-collector · gitlab," (last access 2024.11.29). [Online]. Available: https://code.europa.eu/EDPS/website-evidence-collector/

[22] ungoogled-software, "Release 121.0.6167.184-1 · ungoogled-software/ungoogled-chromium · github," (last access 2024.11.29). [Online]. Available: https://github.com/ungoogled-software/ungoogled-chromium/releases/tag/121.0.6167.184-1

[23] Mathias Bynens, "Github - puppeteer/puppeteer: Javascript api for chrome and firefox," (last access 2024.11.29). [Online]. Available: https://github.com/puppeteer/puppeteer

[24] Disconnect Inc., "Github - disconnectme/disconnect-tracking-protection: Canonical repository for the disconnect services file," (last access 2024.11.29). [Online]. Available: https://github.com/disconnectme/disconnect-tracking-protection

[25] Fanboy, MonztA, Khrin, Yuki2718, and piquark6046, "Github - easylist/easylist: Easylist filter subscription (easylist, easyprivacy, easylist cookie, fanboy's social/annoyances/notifications blocking list):," (last access 2024.11.29). [Online]. Available: https://github.com/easylist/easylist

[26] E. Casey, "Error, uncertainty and loss in digital evidence," *International Journal of Digital Evidence*, vol. 1, no. 2, pp. 1–45, 2002.

[27] K. Inman and N. Rudin, *Principles and Practises of Criminalistics: The Profession of Forensic Science*. Boca Raton Florida, USA: CRC Press LLC, 2001.

[28] Dataskydd.net Sverige, "dataskydd.net/webbkoll-backend: Express.js app that runs puppeteer as a service; visits specified url with chromium and sends back various data (requests, cookies, etc.) as json. - codeberg.org," (last access 22/11/2024). [Online]. Available: https://codeberg.org/dataskydd.net/webbkoll-backend

[29] Oracle Inc., "Oracle VM virtualbox," (last access 2024.11.29). [Online]. Available: https://www.virtualbox.org

[30] ungoogled-chromium Authors, "GitHub - ungoogled-software/ungoogled-chromium: Google Chromium, sans integration with Google," (last access 2024.11.29). [Online]. Available: https://github.com/ungoogled-software/ungoogled-chromium

[31] S. Kiltz, N. Weiler, T.-F. Riechard, R. Altschaffel, and J. Dittmann, "Securware-journal-download-folder," (last access 2024.11.29). [Online]. Available: https://cloud.ovgu.de/s/RWEHi9wSqH3xbjQ

[32] Daniel Stenberg, "curl - command line tool and library for transferring data with urls (since 1998)," (last access 2024.11.29). [Online]. Available: https://curl.se/

[33] Philippe Biondi, "Scapy," (last access 2024.11.29). [Online]. Available: https://scapy.net/

Key Establishment for Maintenance with Machine to Machine Communication in Transportation: Security Process and Mitigation Measures

Sibylle Fröschle
Institute for Secure Cyber-Physical Systems
Hamburg University of Technology
Hamburg, Germany
sibylle.froeschle(at)tuhh.de

Martin Kubisch
Airbus CRT
Munich, Germany
martin.kubisch(at)airbus.com

***Abstract*—Machine to machine communication over wireless networks is increasingly adopted to improve service and maintenance processes in transportation, e.g., at airports, ports, and automotive service stations. This brings with it the challenge of how to set up a session key so that the communication can be cryptographically secured. While there is a vast design space of key establishment methods available, there is a lack of process of how to engineer a solution while considering both security and safety: how to assess the threats and risks that come with a particular key establishment method? And how to iteratively refine a key establishment method under development such that risk is mitigated to an acceptable level? In this paper, we put forward an approach that addresses these questions. Moreover, we devise several cyber-physical measures that can be added to mitigate risk. We illustrate our approach and the mitigation measures by means of a real-world use case: TAGA — a Touch and Go Assistant in the Aerospace Domain. Finally, we highlight the crucial role that simulation has to play in this security process for safety.**

Index Terms*— *Security; Key Establishment; Threat and Risk Analysis; Simulation; Transportation.

I. Introduction

Machine to Machine (M2M) communication over wireless networks is increasingly adopted to improve service and maintenance processes in transportation, e.g., at airports, ports, and automotive service stations. This does not come without security challenges: often these processes are safety-critical, and often, attacks against them would disrupt critical infrastructures. One example are the ground processes at an airport. When an aircraft has landed and reached its parking slot at the apron many processes such as refuelling and pre-conditioning are performed. M2M communication between the aircraft and the respective ground unit allow us to optimize these processes with respect to accuracy of service, energy-efficiency, safety, and time. The aircraft will send sensor values (e.g., temperature or fuel readings), and the ground unit can adopt flow parameters accordingly. However, if an attacker managed to spoof fake sensor values into the M2M communication then this could compromise safety.

The adoption of M2M communication brings with it the challenge of how to set up a session key so that the communication can be cryptographically secured. While there is a vast design space of key establishment methods available, there is a lack of process of how to engineer a solution while considering both security and safety: how to assess the threats and risks that come with a particular key establishment method? And how to iteratively refine a key establishment method under development such that risk is mitigated to an acceptable level? In this paper we address these questions building on our conference contribution [1]. We motivate and illustrate our approach by means of a real-world use case: TAGA — a Touch and Go Assistant in the Aerospace Domain. TAGA is currently under development to enable the introduction of M2M communication for ground processes at airports.

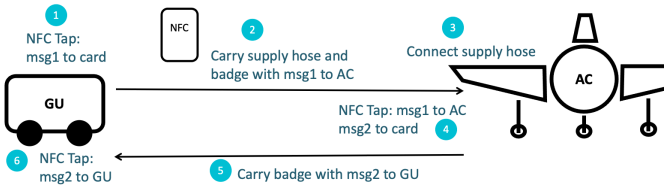


Fig. 1. Pairing up a ground unit and an airplane

The idea behind TAGA is to set up the session key by means of a Near Field Communication (NFC) system. Each aircraft and ground unit is equipped with a TAGA controller that contains a secure element for cryptographic operations and an NFC reader. Moreover, the operator of each ground unit is provided with a passive NFC card. Altogether, this allows them to transport messages for key establishment from the ground unit to the aircraft, and back by means of taps with the NFC card against the respective NFC reader. The 'TAGA walk' can conveniently be integrated into the operator's usual path to the aircraft and back while connecting up the respective supply hose. This is illustrated in Figure 1.

To start off with, TAGA only defines a process of how to transfer the messages of a two-way key establishment (KE) protocol, and there is still considerable design space: which

concrete KE protocol shall we employ? Do we need to add further measures to mitigate risk, and if so which ones?

Integrating M2M communication in transportation has to undergo a safety and security engineering process conform to the safety and security norms applicable to the respective domain (such as ISO/SAE 21434 for road vehicles and DO-178C, DO-254, DO-326A and ARP4754 in the aeronautics domain). This process will typically involve the following activities. First, vulnerable assets have to be identified (such as here the communication channel). Second, for each asset the potential threats have to be collected (e.g., by a keyword-guided method such as STRIDE). And third, for each threat a risk level has to be determined. The risk level is typically determined by, on the one hand, rating the safety impact of the threat, and, on the other hand, rating the likelihood that the threat can be implemented. As a result, the risk level will decide whether protection by security controls is required, and to which assurance level the corresponding security requirements have to be validated.

When it comes to integrating security controls and security systems the most relevant and widely adopted standard is Common Criteria (CC) (ISO/IEC 15408). It is the standard that is widely adopted to evaluate security products and systems. This standard allows us to define a profile of security requirements for a target of evaluation that fall into security functional requirements, and assurance requirements. The latter specify that the security functional requiremens must be validated to a sufficient assurance level. While a CC profile provides a clear interface between safety and security this should not be taken as an excuse to stop short of a stronger integration between security and safety engineering. Without it important safety measures that can mitigate security risks might be overlooked.

Problem and Contribution: While there is a vast design space of key establishment methods and products available, some of them with CC evaluation, there is a lack of process of how to engineer a solution while integrating both security and safety: how to assess the threats and risks that come with a particular key establishment method in a specific context? And how to iteratively refine a key establishment method under development such that risk is mitigated to an acceptable level? In this paper, we put forward and illustrate an approach that addresses these questions.

We proceed as follows. In Section II we motivate and present our overall approach. Our approach is based on the concept of *connection compromise states*, which define how key establishment can fail, and provide a finer-grained interface between security and safety. In Section III we motivate and illustrate our approach by means of the TAGA use case. In Section IV we devise several concrete measures that can be added to mitigate risk. In Section V we give a workflow on how to assess and mitigate the safety impact starting from the connection compromise states. In particular, we highlight the important role of simulation in this workflow. In Section VI we put our work in context with related work. In Section VII we draw conclusions and discuss future work.

This paper extends the conference version [1] as follows. In Section II we additionally provide the rationale behind the connection compromise states. In Section III-D we discuss an intricate consequence of long-term key compromises, and in Section IV we introduce several new mitigation measures. Some of the material is based on the preprint [2].

II. Key Establishment for Vehicle to Service Unit Communication

Setting: We first define the problem setting. As shown by example in Figure 1 we assume that there is a vehicle V that is to undergo a maintenance procedure at some location. The maintenance procedure can involve several types of services, and each service involves at least one service unit. Each service unit is either directly coupled to the vehicle (e.g., via a supply hose) or indirectly (e.g., via the loading of goods). To optimize the maintenance procedure each service unit shall be able to engage in M2M communication with the vehicle it services: to exchange data such as sensor and status values or even instructions on how to move. Several such procedures can take place in parallel in adjacent or remote locations.

We assume that the communication is conducted over a wireless channel (such as Wi-Fi IEEE 802.11), and that a protocol that allows two parties to communicate securely, given a secure session key is already in place. This involves an AEAD (Authenticated Encryption with Associated Data) scheme such as AES-GCM, and measures against replay and reflection such as counters and directionality differentiation (c.f. [3], Section 5.4). For the Wi-Fi security protocols WPA2/3 this is provided by the subprotocol that is responsible for the bulk data handling after the 4-way handshake. Here we focus on the challenge of how to establish the necessary session key between a service unit and the vehicle.

Security Requirements: Table I shows the security properties that any key establishment method for Vehicle to Service Unit (V2SU) communication must at least satisfy. Properties (1) and (2) ensure that the key remains secret, and that it is fresh for each session. Properties (3) and (4) are derived from the standard authentication properties for key establishment protocols [4]. We have formulated the properties without explicitly referring to the names of the peers. This is to allow for *secure device pairing* as the key establishment method of choice, where identities do not necessarily have to be exchanged. Names can, however, be included in the parameter list. One can also include the type of service, and other service specific parameters into the parameter list. Property (5) is specific to our setting: it ensures that the cyber channel indeed connects the machines that are physically coupled in the maintenance service.

Design Space: The state of the art of key establishment offers two approaches to achieve the secrecy and authentication properties: one is to employ an *Authenticated Key Establishment (AKE) Protocol* [5]; the second is to make use of a *Secure Device Pairing (SDP) scheme* [6]. As we will see later a combination is also possible.

AKE protocols [5] are by now well-investigated, and there exist many standardized protocols that come with formal

TABLE I
SECURITY REQUIREMENTS FOR V2SU KEY ESTABLISHMENT

1)	*Secrecy of the session key.* Upon completion of the key establishment method, the service unit and the vehicle should have established a session key which is known to the vehicle and service unit only.
2)	*Uniqueness of the session key.* Each run of the key establishment method should produce distinct, independent session keys.
3)	*Service unit authentication.* Upon completion of the key establishment method, if a vehicle believes it is communicating with a service unit on the session with key k and parameters $p_1, \ldots, p_n$ then there is indeed an authentic service unit that is executing a session with key k and parameters $p_1, \ldots, p_n$.
4)	*Vehicle authentication.* Upon completion of the key establishment method, if a service unit believes it is communicating with a vehicle on the session with key k and parameters $p_1, \ldots, p_n$ then there is indeed an authentic vehicle that is executing a session with key k and parameters $p_1, \ldots, p_n$.
5)	*Agreement with physical setup.* Upon completion of the key establishment method, the service unit and vehicle should also be linked by the respective physical setup.

TABLE II
CONNECTION COMPROMISE STATES FOLLOWING A BREACH OF V2SU KEY ESTABLISHMENT

1)	*Person-in-the-middle (PitM).* The service unit has a connection secured by session key K and the vehicle has a connection secured by key K' but the attacker knows both K and K'.
2)	*Impersonation to service unit (Imp2SU).* The service unit has a connection secured by session key K but the attacker knows K.
3)	*Impersonation to vehicle (Imp2V).* The vehicle has a connection secured by session key K but the attacker knows K.
4)	*Parameter mismatch (ParsMismatch).* A peer has a connection secured by session key K and for a session with parameters $p_1, \ldots, p_n$, and another peer has a connection secured also by K and for a session with parameters $p'_1, \ldots, p'_n$, and the attacker does not know K, but there is $i \in [1, n]$ such that $p_i \neq p'_i$.
5)	*Mismatch with physical setup (PhysMismatch).* A peer P shares a connection secured by session key K with another peer P', and the attacker does not know K, but P and P' are not linked by the respective physical setup.

security proofs. One example is the handshake protocol of Transport Layer Security (TLS). The advantage of AKE protocols is that they are designed to be secure in the presence of active adversaries: their security proofs assume an attacker who has complete control of the network. The drawback is that communication partners need to pre-share a security context such as a pre-shared long-term secret or a public key infrastructure. This typically results in a key management overhead, which can in turn be the source of further threats to the system.

SDP [6] schemes make do without a pre-shared security context but instead rely on so-called Out-of-Band (OoB) channels to safeguard against person-in-the-middle (PitM) attacks. These schemes have been widely adopted for Internet of Things (IoT) and personal devices. One example is Bluetooth pairing of a device to one's smartphone. Often the human user is used as the OoB channel; other schemes make use of properties of wireless channels such as Near Field Communication (NFC). The challenge is that the OoB channel must provide authenticity, and it is not always possible to validate this to a high assurance level: e.g., because a human user is involved or because it is difficult to establish that the wireless channel indeed satisfies authenticity. The great advantage of SDP in our context is that it makes do without a pre-established security context. Moreover, it will help us to achieve Property (5): to pair up two devices typically comes with proximity or some physical interaction, and in our context this can be woven into the procedure of the physical setup of the two machines.

Security Engineering for Safety — Challenge: How to assess the threats and risks that come with a particular key establishment method in our context? And how to iteratively refine a key establishment method under development such that risk is mitigated to an acceptable level? At first sight, one might be tempted to proceed as follows: assess the safety impact when the key establishment method maximally fails (i.e., when the attacker has full control over the connection); derive a safety level, and translate this into a Common Criteria security assurance level; hand this over to a company that provides key establishment products; and acquire a product with the corresponding Common Criteria certificate.

However, this approach has the drawback that it closes the door to measures on the cyber-physical service itself, and hence, to measures that mitigate the safety impact directly. Moreover, in our context where actors come from different security domains we cannot exclude insider attacks, and hence, this approach might overlook some threats that cannot be reduced in their likelihood by even the highest assurance level.

Connection Compromise States: Instead, we wish to reflect that a successful attack against a key establishment method can have different outcomes, and that certain outcomes might be easier to achieve for the attacker than others. To this end, we identify in which ways a supposedly secure connection can be compromised following a breach of the key establishment method. The resulting *connection compromise states*[1] are described in Table II and illustrated in Figure 2.

We now explain the rationale behind the connection compromise states. Assume a vehicle V is undergoing maintenance at some location L. We explore how key establishment could have failed from the view of V, and from the view of a service unit at L respectively. Figure 3 shows the derivation from the view of a service unit U at L. The derivation from the view of V is similar.

Assume that U has established a session key K, supposedly with V. In the left branch of Figure 3 we consider the case when secrecy of K has been breached. Then the attacker knows K, and will be able to run the connection with U impersonating V. Hence, the attacker has reached the connection compromise state *impersonation to service unit (Imp2SU)*.

Next we ask whether V has established a key K' for S_U, the service provided by U (where the case $K' = K$ is included).

[1]It is important to note that here we only consider compromise states directly derived from a failure of the key establishment goals. Other compromise states, e.g., such as those that result from attacks against session management such as session hijacking, fixation or riding are out of our scope.

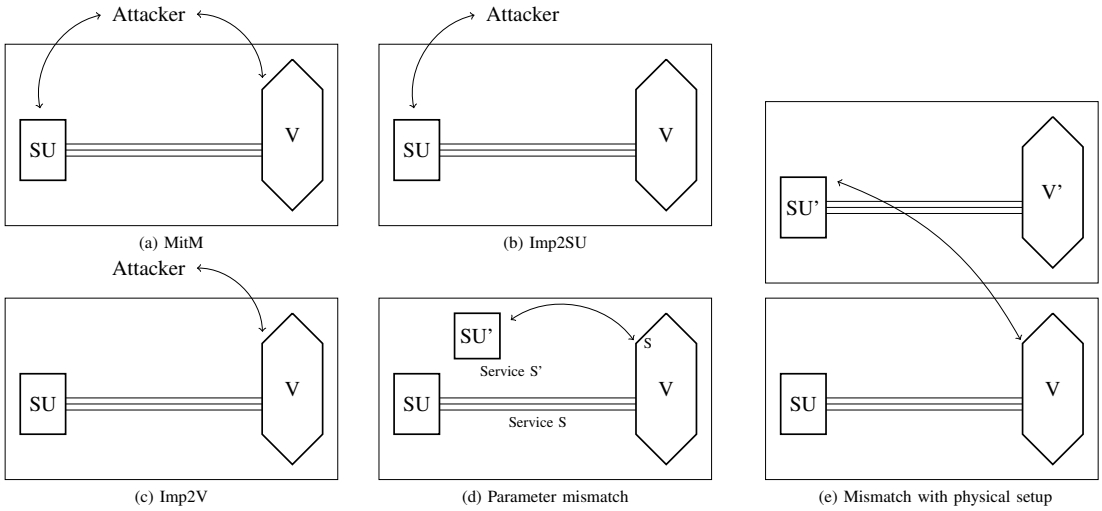


Fig. 2. Illustration of the connection compromise states (resulting from a failure of key establishment[1])

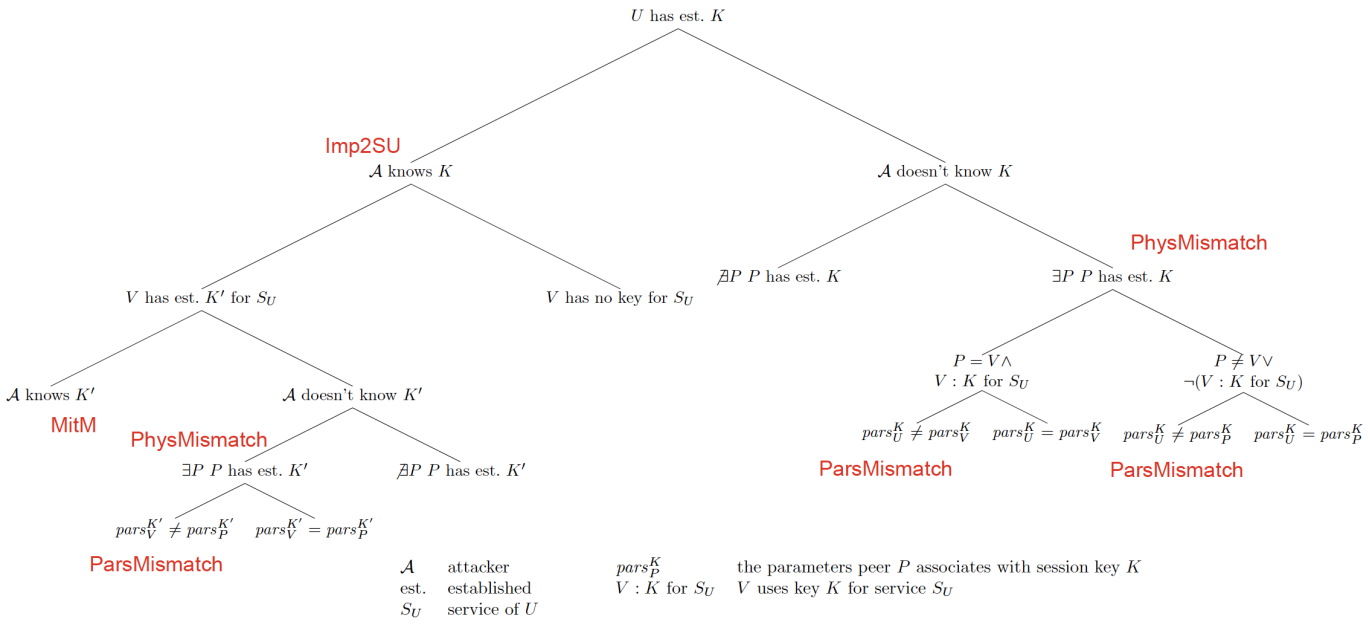


Fig. 3. Deriving the connection compromise states for the view of service units

If this is so, and secrecy of K' is also breached then, in addition, the attacker will be able to impersonate U to V. Hence, the attacker has full control over the communication between V and U: they have reached the *person-in-the-middle (PitM)* connection compromise state.

If V has established a key K' for S_U but the attacker does not know K' then either V must have established the key with a peer other than U, say P, or there is no peer who has currently established the key K'. The latter case can be avoided by U itself when we ensure that the key establishment method contains a freshness proof and a key confirmation step.

The first case when such a peer P exists brings with it a violation of agreement with physical setup: U is or will be physically linked to V rather than P. Hence, we have reached a *mismatch with physical setup* connection compromise state. Moreover, this can go along with a violation of authentication in that there is no agreement with one of the parameters. Then, in addition, we have reached a *parameter mismatch* state.

We still need to consider the case when V has no key established for S_U. Since this case does not directly influence the service S_U we will not expand this further here. This case is covered by the analogous derivation from the view of V.

Note that while it is often the attacker's best strategy to breach vehicle authentication, service unit authentication, or both respectively in order to reach Imp2SU, Imp2V, or PitM respectively, it is not necessary to do so: e.g., the key

could have been revealed after a successful run of the key establishment method between two authentic parties.

Let us now turn to the top right branch of Figure 3, and explore how key establishment could have failed while secrecy of K is *not* breached. Then either there exists a peer P that has established K (right branch) or not (left branch). Similarly to above, the latter case can be avoided by U itself when we ensure that the key establishment method contains a freshness proof and a key confirmation step. In the first case, we ask whether P is indeed V, and V indeed uses K for service S_U. If this is so then there can at most be a *parameter mismatch*. If this is not so then we have a *mismatch with physical setup*. Moreover, this can also go along with a *parameter mismatch*. To reach a parameter mismatch state the attacker always has to breach authentication. Clearly, a mismatch with physical setup state always breaches agreement with physical setup.

Security Engineering for Safety — Approach: The security engineering activities can now be carried out in a structured and systematic fashion as follows:

1) The security experts identify the threats against the key establishment method under investigation, and assess for each connection compromise state the likelihood that this state can be reached by an attacker.
2) The safety and process engineers of the vehicle and the maintenance procedure assess for each connection compromise state what the severity of impact on safety (and perhaps other factors) will be if the attacker manages to reach this state. Moreover, they explore whether and how the impact can be mitigated by process measures.
3) At synchronization points safety and security experts together decide whether the combination of the current assessments of threat likelihood and safety impact result in an acceptable risk level. If not the workflow will be repeated in an iterative fashion until an optimal solution is reached. Finally, assurance levels for the security components and the mitigation safety measures will be derived, and forwarded for development, or product integration respectively.

We will discuss a workflow for the activities of Part (2) in more detail in Section V since this is where simulation plays a crucial role throughout. Part (1) will be illustrated via our case study. Here simulation might also play an important role, e.g., to analyse channel properties with respect to a SDP scheme. For a detailed analysis we employ the tools for formal protocol verification, such as the Tamarin Protocol Verifier [7].

III. TAGA: A Touch and Go Assistant in the Aerospace Domain

We now illustrate our approach by means of the real-world use case TAGA.

A. Preliminaries

In the following, we will make use of the basic Diffie-Hellman exchange as well as authenticated Diffie-Hellman protocols. We assume a cyclic group G of prime order n, and a generator P of G such that the decisional Diffie-Hellman problem is hard in G. The domain parameters G, n, and P can be fixed or sent as part of the first message. We use small letters to denote elements of the field $\mathbb{Z}_n^*$, and capital letters for elements of G. A key pair in the protocols consists of a public key T, which is a group element, and a private key t, which is an element of the field $\mathbb{Z}_n^*$ such that $T = tP$. Group operations are written additively ($A + B$, and cA) consistent with notation for elliptic curve cryptography.

TABLE III
Protocol notation

G	ID of ground unit
A	ID of aircraft
S	Service name of ground unit
L	Location (i.e., parking slot) of the process
I	Intruder
$ssid_A$	SSID of the aircraft's WLAN
$ssid_I$	SSID of the intruder's WLAN
R_X, r_X	Ephemeral public and private DH key of party X
W_X, w_X	Long-term public and private DH key of party X
mac	A message authentication code algorithm
H	A cryptographic hash function
KDF_1, KDF_2	Key derivation functions
K	Resulting session key
K'	Derived mac key
$m_1 \| \| m_2$	The concatenation of messages m_1 and m_2

To describe the protocols we use the notation presented in Table III. Moreover, we use DH key short for Diffie-Hellman key, GU short for ground unit, AC short for aircraft, and OP short for Operator.

B. The TAGA Protoype

The TAGA Pairing Process: The prototype of TAGA pairing is based on an unauthenticated three-pass key establishment protocol, where the third pass is a key confirmation step. It is illustrated in Figure 4 for the case when the Diffie-Hellman key exchange is used as the underlying protocol.

The operator performs a first NFC tap at the ground unit. Thereby a first message M_1 is written to the card. M_1 contains information necessary for establishing the key together with the ID of the ground unit and the service that it provides. Then the operator walks to the aircraft. Typically they will also carry a supply hose; e.g., for pre-conditioning they will carry the air supply hose.

At the aircraft, the operator first performs some physical setup, such as connecting the supply hose to the supply port, and then carries out the second NFC tap. Thereby, M_1 is transferred to the aircraft's TAGA controller, and a second message M_2 is written onto the card. M_2 contains information necessary for establishing the key together with the ID of the aircraft and access data to its WLAN such as the SSID. M_2 also contains a ciphertext to grant key confirmation to the ground unit. The operator then walks back to the ground unit.

Back at the ground unit, the operator carries out a final NFC tap, and transfers M_2 to the ground unit's TAGA controller. The ground unit is now able to connect to the aircraft's WLAN. A third message is passed over the WLAN connection to achieve key confirmation to the aircraft. Finally, the operator

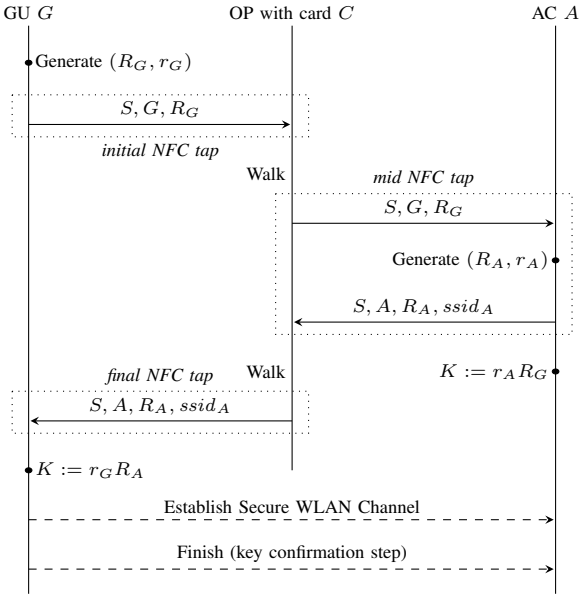


Fig. 4. TAGA pairing with Diffie-Hellman key exchange

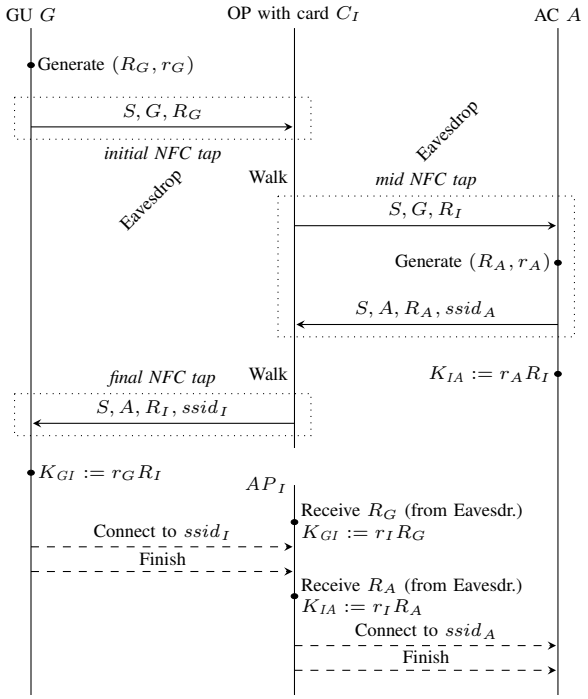


Fig. 5. Person-in-the-middle attack by card swapping and eavesdropping

activates the ground unit; e.g., for pre-conditioning they switch on the air supply. Now the ground unit and the aircraft are ready to carry out the service using M2M communication.

Threats against the TAGA Channel: Even though TAGA takes place in a secure zone, where only authorized personnel have access, our analysis has shown that there are many indirect ways of compromising the TAGA channel. One example is that the attacker might swap a counterfeit card for the TAGA card, e.g., while the operator takes a break. Another example is that the attacker might eavesdrop on the NFC exchange from outside the secure zone of the turnaround, e.g., by using a special antenna to increase the nominal range of NFC.

The following example shows that the combination of card swapping and eavesdropping already allows the attacker to implement the classic person-in-the-middle attack against the basic Diffie-Hellman exchange over the TAGA channel.

Example 1 (PitM by Swap & Eavesdrop). Let A be an aircraft and G be a ground unit at parking slot L so that G is to service A. In preparation, the attacker swaps his own prepped card C_I for the operator's card, e.g., while the operator is on a break. Moreover, the attacker sets up NFC eavesdropping capability, and their own WLAN access point AP_I in the range of L. Both C_I and AP_I are prepped with a fixed DH key pair (r_I, R_I), and the SSID $ssid_I$ of the attacker's WLAN.

The attack then proceeds as depicted in Figure 5. The card C_I carries out the first tap as usual. However, with the second tap the counterfeit card writes the attacker's public key R_I to A rather than G's public key R_G. Similarly, with the third tap the card writes R_I and $ssid_I$ to G rather than A's public key R_A and SSID $ssid_A$. Hence, G computes session key K_{GI} based on r_G and R_I, and A computes session key K_{IA} based on r_A and R_I.

To be able to compute the same keys the attacker needs to get R_G and R_A onto their access point AP_I. Even if the card only has a passive NFC interface they can use eavesdropping to do so. Once they have computed K_{GI} and K_{IA} they can establish the corresponding channels, and mount a PitM attack against the M2M communication between G and A.

Estimating the Safety Impact: To estimate the severity of impact of a PitM connection compromise we consider the two ground services fuelling and pre-conditioning. Our examples show that while for fuelling the safety impact is controlled by inbuilt safety measures this is not the case for pre-conditioning, and the safety impact is potentially high.

Example 2 (Fuelling). The attacker can forge fuel orders, and induce the fuel truck to load an insufficient or surplus amount of fuel. While this can be highly disruptive there is no safety impact. Since the aircraft measures the fuel itself it will notice if the loaded fuel is not sufficient. Moreover, if the attacker tries to cause spillage (and hence, a fire hazard) by too large a fuel order this will not succeed since the backflow will stop the pump of the fuel truck.

Example 3 (Pre-Conditioning). The attacker can forge airflow parameters and sensor values that will induce the pre-conditioning unit to apply air pressure and temperature unsuitable to the aircraft. This can be highly damaging: if the cooling process is too fast then water in the pipes can quickly become frozen and clog up the pipes. This can happen very quickly: e.g., with the lowest inlet temperature within 30 seconds, with safety considerations still within 100 seconds. The resulting backflow will be detected by the pre-conditioning unit. However, in the worst case pipes might already have burst. In any case the pipes have to be checked for damage

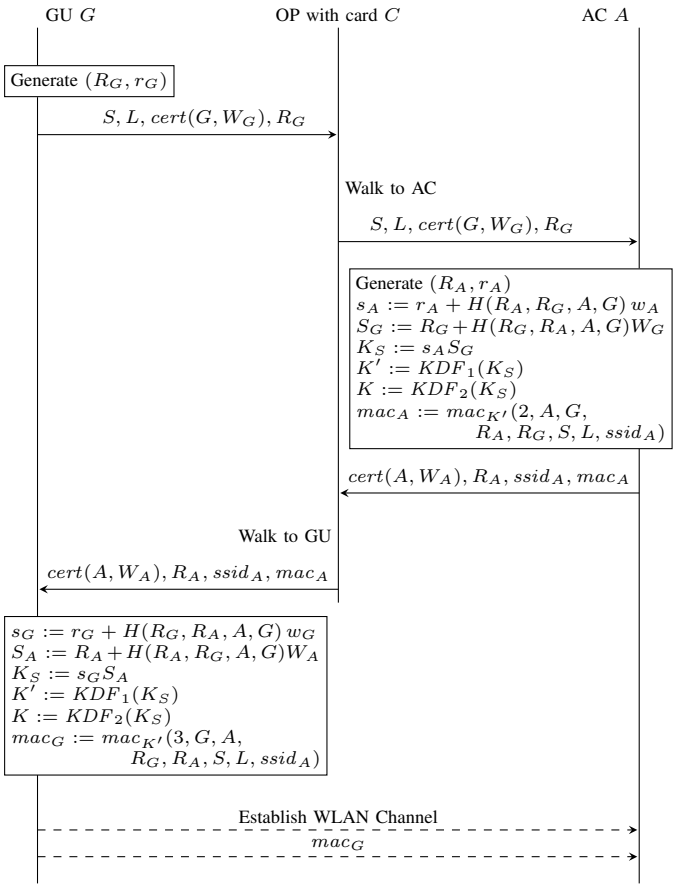


Fig. 6. TAGA pairing based on the FHMQV protocol

afterwards, which is a costly procedure.

In the worst case, the attacker could try to optimize the attack based on the sensor values sent by the aircraft: they could try to control the airflow in a way that maximizes the strain on the pipes without this being detected during service time but with a high risk that pipes burst during flight.

Our analysis of the prototype has shown that one either needs to refine TAGA by better protecting the TAGA channel, or by using an AKE protocol instead of the basic Diffie-Hellman exchange. In the following, we illustrate aspects of the latter refinement. A solution in line with the first refinement can be found in [8].

C. Refinement: Authenticated TAGA

The Authenticated Setting: In the setting of authenticated TAGA, every aircraft A has a long-term key pair (W_A, w_A), where W_A is the public key and w_A is the private key. Moreover, A holds a certificate for its public key W_A, which is issued by the airline $\mathcal{A}$ that owns A (or an entity commissioned by $\mathcal{A}$). We denote the certificate by $cert_{\mathcal{A}}(A, W_A, T_A, V_A)$, where T_A is the aircraft type of A, and V_A specifies the validity period of the certificate.

Analogously, every ground unit G has a long-term key pair (W_G, w_G), and a certificate for its public key W_G, which is issued by the airport $\mathcal{H}$ that harbours G (or an entity commissioned by $\mathcal{H}$). We denote the certificate by $cert_{\mathcal{H}}(G, W_G, S_G, V_G)$, where S_G is the service type of G and V_G is the validity period of the certificate.

We assume that every aircraft has installed the root certificates of those airports it intends to land at, and each ground unit has installed the root certificates of those airlines it is authorized to handle. For short notation, we often write a certificate $cert_{\mathcal{A}}(A, W_A, T_A, V_A)$ as $cert(A, W_A)$ when the issueing party, type of aircraft or service, and validity period are implicitly clear from the context. Similarly, we often write $cert(G, W_G)$ short for $cert_{\mathcal{H}}(G, W_G, S_G, V_G)$.

Figure 6 shows TAGA based on the *Fully Hashed Menezes-Qu-Vanstone protocol (FHMQV)* [9], [10], where for TAGA we include service and location into the key confirmation step. FHMQV is one of the strongest protocols regarding security, resilience and efficiency, and comes with a security proof. It satisfies all our secrecy and authentication requirements, i.e., Properties (1)–(4) of Table I, even when assuming that the attacker has full control of the TAGA channel. Our requirement 'Agreement with physical setup', i.e., Property (5), can also be guaranteed. Since we have included the parameters service and location into the key confirmation step the ground unit and aircraft will agree on service and location as part of the authentication guarantees. Then to obtain Property (5) the aircraft and ground unit only need to carry out a handshake of 'ready for service' messages once the secure channel is established.

The Threat of Long-Term Key Compromise: While secure AKE protocols are designed to withstand an attacker who has full control of the network they are vulnerable to the threat of *long-term key compromises*. We say the attacker has obtained a *long-term key compromise (LTKC)* of the aircraft A if they have managed to get hold of credentials that authenticate A: a public/private key pair (W_A, w_A) and a valid certificate $cert(A, W_A)$, which asserts that W_A belongs to A. The definition for a ground unit G is analogous.

Given the LTKC of a party P, it is unavoidable that the attacker can impersonate P to other parties. In classical settings of AKE protocols this will typically impact on the resources of P, and only P, itself. However, in our setting, a LTKC can have a wider impact. The following example shows how the attacker can use the LTKC of some aircraft A_I (possibly of an airline with key management of low security quality) to impersonate A_I to a ground unit that is physically connected to another aircraft A (possibly of an airline with key management of high security quality).

Example 4 (Impersonation to Ground Unit with LTKC of any Aircraft). Let A_I be a real or non-existent aircraft of airline $\mathcal{A}_I$, and assume that the attacker has achieved a LTKC of A_I. Further, let A be an aircraft of airline $\mathcal{A}$, and G be a ground unit at airport $\mathcal{H}$ such that G provides service S to A during turnaround at parking slot L. In preparation, the attacker swaps their own counterfeit card C_I for the card of G's operator. Moreover, the attacker sets up NFC eavesdropping capability, and their own WLAN access point AP_I within range of L.

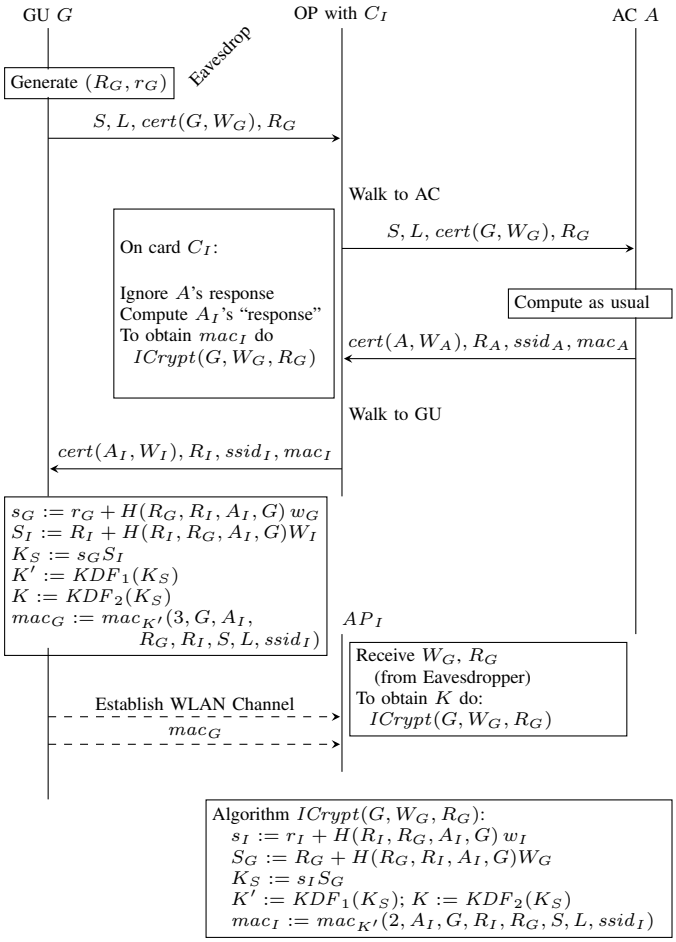


Fig. 7. Impersonation to ground unit with LTKC of any aircraft

Both AP_I and C_I are prepped with A_I's long-term credentials w_I and $cert(A_I, W_I)$, a fixed ephemeral key pair (r_I, R_I), and the SSID $ssid_I$ of the attacker's WLAN.

Then the attacker can proceed as shown in Figure 7: they simply establish a key with G using A_I's credentials rather than those of A. Since A_I's ephemeral key pair can be fixed beforehand, the resulting session key can be computed independently on the card C_I, and the attacker's WLAN point AP_I respectively. The latter only needs to receive G's public keys by relay from the eavesdropping device.

Estimating the Safety Impact: The attacker has only obtained an Imp2SU connection compromise, and one may hope that this comes with less safety impact than PitM. However, Imp2SU still allows the attacker to feed any sensor values they like to the ground unit while the ground unit thinks this information stems from the aircraft and adjusts the service correspondingly. The safety impact is potentially high for pre-conditioning.

Example 5 (Pre-Conditioning). The attacker feeds in airflow parameters and sensor values, and the ground unit will control the airflow based on this information. Since the air supply leads directly into the mixer unit of the aircraft this will take immediate effect without the aircraft itself having to open a valve or the like first. Crew or ground staff might notice that something is wrong and switch off the air supply manually. However, as explained in Example 3 damage can occur quickly and this might be too late. In contrast to the PitM attack, the attacker is not able to obtain sensor values sent by the aircraft, and, hence, they are not able to optimize the attack based on such information.

Given the potential safety impact and scale of the attack (given one LTKC of any airline) it is clear that a further refinement of the TAGA method is necessary. In particular, it is worth exploring measures that work on the ground service itself: one airline will not have much control over the security infrastructures managed by another. In addition, in our context of critical infrastructures one cannot write off that a state actor might take influence to obtain and abuse valid aircraft credentials of an airline in its realm. We will propose several measures that will address this situation in Section IV.

Given the LTKC of a ground unit, a simple check can ensure that the compromised credentials cannot be employed by the attacker beyond the realm of the airport where the ground unit operates: the aircraft can simply check the airport in the certificate against its current location. Moreover, when the physical control of the service lies entirely with the ground unit then an Imp2V attack is usually less harmful.

D. Intricate Consequences of LTKCs: Key-Compromise Impersonation

Given that a participant X has a LTKC, it is clear that the attacker can impersonate X to any other particpant. And this is what we have considered so far. However, a more intricate question to ask is whether this enables the attacker to impersonate any other participant to X. We then say the attacker can carry out a *Key-Compromise Impersonation (KCI)* attack [11]. In our setting this would mean: given a LTKC of ground unit G_I, the attacker will be able to stage an Imp2SU attack against any aircraft serviced by G_I. Moreover, the attacker can combine each such KCI attack with a standard impersonation attack to obtain PitM capability.

Fortunately, many AKE protocols such as the FHMQV used here are resilient against KCI attacks. And hence, this attack with potentially large-scale impact can be excluded by choice of the protocol. We illustrate the KCI attack by a concrete example based on the *Unified Model (UM)* protocol (c.f. [12]). The UM, shown in Table IV, is another Diffie-Hellman protocol with implicit authentication. Moreover, it is well-known to be vulnerable to KCI attacks [12].

Example 6 (KCI Attack against UM). Let G_I be a ground unit for service S at airport $\mathcal{H}$, for which the attacker has achieved a LTKC. Further, let A be any aircraft that is serviced by G_I, say at parking slot L. In preparation, the attacker swaps the NFC card of G_I's operator with their own prepared card U_I. Moreover, they set up NFC eavesdropping capability, and their own WLAN access point AP_I within range of L. Both, AP_I and the card U_I, are prepped with

TABLE IV
THE UM PROTOCOL

1) G generates (R_G, r_G)
   G sends $S, L, cert(G, W_G), R_G$
2) A receives and validates the message
   A generates (R_A, r_A)
   A computes $K := H(w_A W_G \,||\, r_A R_G)$
   A computes $mac_A := mac_K(2, A, G, R_A, R_G, S, L, ssid_A)$
   A sends $cert(A, W_A), R_A, ssid_A, mac_A$
3) G receives and validates the message
   G computes $K := H(w_G W_A \,||\, r_G R_A)$
   G validates mac_A
   G computes $mac_G := mac_K(3, G, A, R_G, R_A, S, L, ssid_A)$
   G establishes the WLAN connection and sends mac_G.

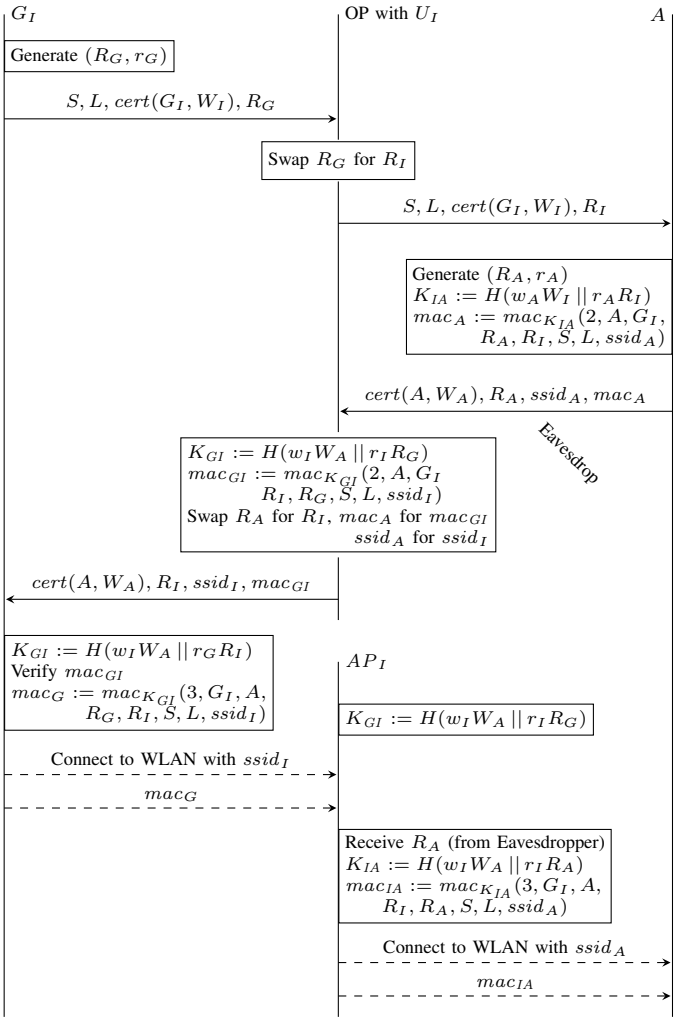


Fig. 8. KCI attack against TAGA with the UM protocol

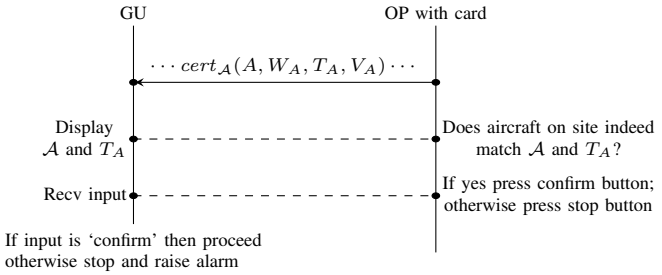


Fig. 9. Last NFC tap extended by human verification of aircraft domain

a fixed ephemeral key pair (r_I, R_I) and the SSID $ssid_I$ for the WLAN with G_I. In addition, AP_I is prepped with G_I's credentials w_I and $cert(G_I, W_I)$, and A's long-term public key W_A. Alternatively, W_A can be obtained by eavesdropping.

Then the attacker can proceed as shown in Figure 8. The interaction with G_I consitutes a KCI attack, where the attacker impersonates A: they can compute the same key K_{GI} as G_I by using their knowledge of w_I rather than w_A (and their own ephemeral key pair). The interaction with A constitutes a standard impersonation attack, where the attacker can impersonate G_I due to their knowledge of w_I, and establishes a key K_{IA} with A. Altogether, the attacker can now fully control the M2M communication of A and G as the PitM.

This illustrates that as long as KCI attacks are a threat it is not possible to protect against one-sided LTKCs by detection measures against impersonation attacks. Hence, in the security analysis it is important that all paths are investigated how an attacker could obtain one of the connection compromise states, even the most intricate ones.

IV. CYBER-PHYSICAL MITIGATION MEASURES

The last section has shown that in a setting where entities of several security domains interact in an ad hoc fashion (so that their digital identities are not known prior to key establishment) the likelihood of certain LTKCs might be comparatively high. We now propose several measures that protect against Imp2SU or even PitM. To allow for a more precise description we formulate most of the measures in the setting of TAGA. Note, however, that it is straightfoward to translate the measures to other settings, and to employ them against Imp2V analogously. A summary is provided in Section IV-E.

A. Human Verification of Aircraft Domain

The likelihood of Imp2SU against aircraft of an airline with high security standards can drastically be reduced if we ensure that the attacker cannot make the ground unit accept a certificate that does not agree with the domain of the aircraft that is actually on site. This can be achieved by a simple measure that makes use of the awareness of the human operator: while the ground unit has no means to verify that the received certificate (and information therein) indeed belongs to the aircraft present at the parking slot, the operator has sight of the aircraft. Hence, they are able to verify that visually

observable features of the aircraft such as its type and airline agree with the information received by the ground unit.

Measure 1 ('Two eyes' verification of aircraft domain (2EV)). Assume the TAGA controller of the ground unit is equipped with a display and two input buttons: one to confirm, and the second to stop the process and raise an alarm. Then the last NFC tap can be extended by human verification as illustrated in Figure 9. First, the operator transfers the second message by NFC tap to the ground unit as usual. Recall that this message contains a certificate $cert_{\mathcal{A}}(A, W_A, T_A, V_A)$, where A is the ID of the aircraft, $\mathcal{A}$ is the airline of A, and T_A is the type of A. Second, the ground unit shows $\mathcal{A}$ and T_A on its display, and the operator verifies whether the aircraft they see on the parking slot is indeed of airline $\mathcal{A}$ and type T_A. If yes, then they will confirm the process; otherwise they will stop the process and raise an alarm.

Unintended errors of the operator can be kept small: they can be trained to keep awareness by injection of false alarms (similary to security screening at airports). It is also possible to implement this with dual control.

Measure 2 ('Four eyes' verification of aircraft domain (4EV)). For increased security a member of the aircraft crew can accompany the ground operator and perform the visual verification as well.

Note that if the underlying protocol is not KCI resilient then the attack shown in Example 6 is possible even when this measure is in place.

B. Time-based Detection

An attacker who carries out an Imp2SU attack in the TAGA setting will need to ensure that the NFC tap at the aircraft looks successful to the operator. Assume we add a 'two eye' verification step in which the operator must verify that the aircraft has indeed received a TAGA request for the service they carry out, and hence starts a respective session. An attacker who only has the capability to reach Imp2SU will not be able to successfully complete the session, and thereby be caught out: the aircraft will raise an alarm when a session is still pending after an unusually long time. Operators can then check what is going on, and, deactivate the ground unit before damage occurs.

For the latter to work it is important that the ground unit defers all safety-critical processing until it can be sure that no alarm will be raised. How long should the ground unit wait for? This can be derived as follows.

Fix a ground service S. Let t^A_{max} be the maximal time that the aircraft waits after starting a new TAGA session for service S to receive the corresponding Finish message. Let t_{min} be the minimal time required from the point after the second NFC tap at the aircraft up to the point when the ground unit has received the final NFC tap. t^A_{max} and t_{min} will mainly be determined by how long the operator needs to walk from the aircraft's TAGA controller back to the ground unit.

Let t^{stop}_{max} be the time the operator maximally needs to carry out an emergency stop at the ground unit once they have heard the alarm of the aircraft. t^{stop}_{max} will include the time the operator will need to walk back to the ground unit from anywhere where they could potentially stay in the meantime.

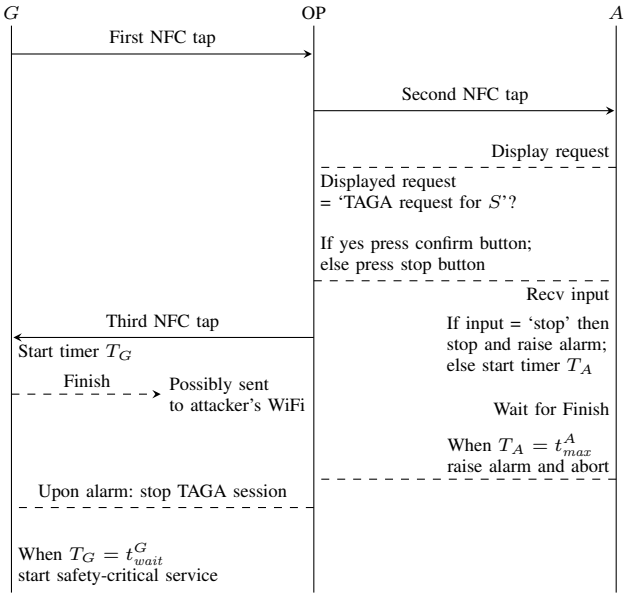


Fig. 10. Time-based detection

Then clearly we have: if the TAGA session of the ground unit has not been stopped within $t^G_{wait} = t^A_{max} - t_{min} + t^{stop}_{max}$ time after it has received the final NFC tap then the local aircraft must have successfully established a session for S, and the ground unit can start with safety-critical processing.

Measure 3 (Time-based detection (TD)). Let S, t^A_{max} and t^G_{wait} be given as above. Assume the TAGA controller of the aircraft is equipped with a display and two input buttons: one to confirm, and the second to stop the process and raise an alarm. Then TAGA can be extended by a time-based detection measure as illustrated in Figure 10. At the second NFC tap, the operator verifies with the help of the display that the request received by the aircraft coincides with a TAGA request for the service S they carry out. If this is confirmed the aircraft will start a timer, say T_A. If the corresponding Finish message is not received before T_A has reached t^A_{max} then the aircraft will raise an alarm. This will trigger the operator to go to the ground unit and stop the unit's service. The ground unit defers any safety-critical processes or settings until t^G_{wait} time has passed from the point of the third NFC tap onwards. If no alarm has been raised and the operator has not stopped the TAGA session by then the ground unit can conclude that no attack has occurred, and continue as usual.

This measure comes with a trade-off between usability and efficiency: if t^A_{max} is set too large then the process takes a lot longer than necessary; if t^A_{max} is set too small then there will be too many false positives and/or pressure on the operator to hurry. t^{stop}_{max} can be chosen to be small when the operator is required to stay close to the ground unit.

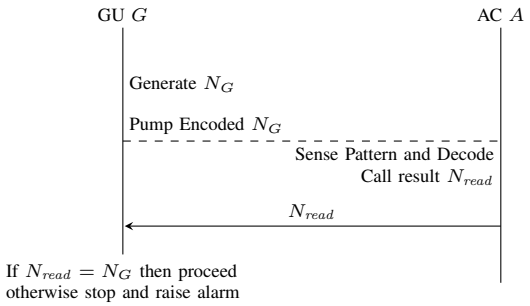


Fig. 11. Physical Challenge/Cyber Response

C. Physical Challenge/Cyber Response

We now propose a measure that translates the standard scheme of challenge/response authentication into the concept of *physical challenge/cyber response:* the ground unit sends a challenge via the physical connection, e.g., encoded in a pattern of pulsating flow, which the aircraft must answer via the cyber channel. Thereby the physical connection is directly bound into the key establishment method.

Measure 4 (Physical Challenge/Cyber Response). Assume that the airpacks of the aircraft are equipped with mass airflow sensors that can detect a pattern of airflow changes and report it to its TAGA controller. Then a phase of physical challenge response can be included before the regular M2M communication starts as illustrated in Figure 11. The ground unit G generates a random number of a fixed size, say N_G, and encodes this into a pattern of pulsating airflow. The aircraft A reads the physical signal using its airflow sensors and decodes it back into a number, say N_{read}. A then responds by sending N_{read} back to G via the cyber channel. G checks whether $N_{read} = N_G$. If this is true then G concludes that it speaks to the aircraft it is physically connected to: only this aircraft could have known N_G. If the numbers don't agree G stops and raises an alarm.

Note that physical challenge/cyber response only counters Imp2SU attacks, and can be undermined by a PitM attacker.

The space of nonces must be sufficiently large to reduce the risk of guessing attacks: even when the attacker cannot receive the physical signal they can always guess the nonce N_G and send it back via a cyber channel they have established with the ground unit by an impersonation attack. This brings about a trade-off between security and efficiency. For example: Say the physical channel allows a binary encoding of numbers in terms of high and low airflow (e.g., using stuffing to synchronize). Say an encoded bit requires 2 seconds to be transmitted, and a challenge shall maximally take 10 (or 20) seconds to be transmitted. Then one can use a space of 32 (or 1024) nonces, and the attacker has a 1/32 (or 1/1024) chance to guess correctly.

D. Attack Detection during M2M Phase

Finally, one can make use of attack detection units that monitor the service during the M2M phase. We present here two examples. However, any attack detection system that detects anomalies falls into this category.

Measure 5 (Safety Check and Safety Alert). The vehicle or service unit could integrate sensors to check whether system variables such as temperature or pressure are about to cross safety limits. Then an alarm could be raised, and operators could deactivate the machine from which the danger emanates. Note that it is not possible to deactivate the machine automatically: it is the machine opposite to the one that raises the alarm that will need to be switched off. Moreover, since the communication channel is thought to be under attack it is not possible to reliably send a deactivation request message to the peer machine either.

Measure 6 (Physics-based Attack Detection). Physics-based attack detection employs a physical model of the normal behaviour of the system to monitor whether real-time measurements of system variables are consistent with the expected behaviour of the system [13], [14]. This concept could be applied in our context as follows. As with the previous measure the vehicle is equipped with sensors that take real-time measurements of system variables. A digital twin of the control of the service unit models the expected behaviour under the assumption that the service unit indeed receives the sensor values the vehicle communicates. If there is a deviation to the actual behaviour then an alarm will be raised. As with the safety check method, it is the opposite machine, here the service unit, that needs to be deactivated, and hence, this has to be carried out by operators.

The advantage of these measures is that they are independent of how key establishment has failed, and also work in the presence of attacks beyond those during key establishment. However, they might not be able to catch the attacks before damage has already occurred: since the physical impact is caused by the opposite machine there is the time delay between the alarm and the operator being able to switch off the service unit. Another challenge is that attacks might go unnoticed if the attacker chooses a stealthy strategy.

E. Summary

In Table V we provide an overview of the measures. Most of them work against Imp2SU but can be masked out if the attacker has full PitM capability or can run a Mismatch attack in parallel (in the case of time-based detection). Measure 1 will detect any attack that makes use of a vehicle certificate from a different domain than the vehicle on site. If the protocol guarantees KCI (Key Compromise Impersonation) resilience [11], [12] then this measure will exclude all Imp2SU attacks.

Measure 1 has the advantage that it is independent of the actual service while Measures 3 and 4 are specific to the service, and might not always be an option. Measure 2 is dependent on the service only in that the time interals t^A_{max} and t^G_{wait} might differ across services, which is straightforward to manage by service-dependent configurations. More challenging might be if it turns out that the time intervals need to be adjusted across airports.

TABLE V
OVERVIEW OF THE MEASURES AND THEIR CHARACTERISTICS

	Measure	Attack	Service-dependent?	Preventive?	Comments
1	2/4-Eyes Verification	Use of non-domain vehicle cert.	no	yes	requires training of operators
2	Time-based Detection	Imp2SU**	configurable	configurable	requires training of operators
3	Physical Challenge/Cyber Response	Imp2SU*	yes	yes	
4	Attack Detection during M2M Phase	all	yes	no	

** ... in the absence of PitM and Mismatch
* ... in the absence of PitM

Measures 1 and 3 are directly bound into the key establishment method, and are therefore preventive in that key establishment will not be successfully completed in the presence of the respective attack. Measure 2 can be implemented in a way so that the attack can be detected before any damage can be caused — in a trade-off with time. Measure 4 might not be able to prevent damage in general but has the advantage that it works for all connection compromise states including attacks that come after key establishment.

V. ASSESSING AND MITIGATING THE SAFETY IMPACT

We now describe a workflow of how the engineers of the maintenance procedure can iteratively assess the severity of impact, and explore and assess means to mitigate it. The workflow consists of the following activities. They can systematically be performed for each of the services, and for each of the relevant connection compromise states. In each of the steps simulation plays a crucial role.

1) Initial estimation and, if applicable, demonstration of the safety impact.
2) Refined analysis of the safety impact.
3) Exploration and assessment of mitigation measures.

Then iterate steps (2) and (3) until risk is mitigated to an acceptable level.

A. Initial Estimation of the Safety Impact

A first analysis of the safety impact is carried out. Usually, this can be done by hand by the engineers of the machines and maintenance process. This gives a first impression of whether a connection compromise state is critical or not. Our examples in Section III show that there can be differences across the services as well as across the connection compromise states.

It makes sense to carry out this initial step breadth-first for all services at hand. In this way one can learn early on if there are large differences between the risk levels across the services. Then one can e.g., partition them into several safety domains, or, mitigate the risk of individual services by additional measures.

Simulation can be an important tool at this stage to demonstrate the safety impact. This should not be underestimated: a demonstration is worth immensely more than a 1000 words when it comes to informing other team members or convincing management of the necessity of security measures (and their costs).

TABLE VI
ATTACKER'S STRATEGIC GOALS

The attacker's strategic goal could be as follows:
1) create maximal damage while the maintenance process takes place,
2) create maximal damage during the operation of the vehicle after the maintenance process has taken place,
3) create maximal disruption, e.g., in terms of delays, equipment cost, locations affected,

while
a) the attack does not remain stealthy,
b) the attack remains stealthy,
c) the attack potential can be demonstrated without being carried out (in view of ransomware attacks).

B. Refined Analysis of the Safety Impact

Many outcomes of the first phase will require a more refined analysis. In the positive case, when the initial estimation has delivered the result that the safety impact is controlled by existing safety mechanisms (c.f. Example 2) it might be important to submit this outcome to closer examination. This is so because safety measures such as backflow valves will not have been desigend to withstand malicicous intent, and the forces or patterns applied might be different when the system is under attack. In the negative case, when the initial estimation has delivered the result that safety impact is to be expected it might be important to explore the attack capabilites in more detail, e.g., to determine whether the attack will only lead to disruption or put passengers at risk (c.f. Example 3).

For this phase we assume that the service under investigation is already modelled in a tool such as Stateflow/Simulink. The model then only needs to be extended to integrate the respective connection compromise state. We suggest to provide one channel component for each of the connection compromise states in addition to the original uncrompromised channel component. Then during evaluation one can switch between the different channel models as required.

The question remains of how to choose the input values for the attack simulations. E.g. to assess the Imp2SU state, which sensor inputs shall the attacker model communicate to the model of the service unit? At first sight, it might seem plausible to use the fault models typically used in safety analysis such as 'stuck at' or 'random'. However, this will not sufficiently reflect that during an attack the values are chosen by a purposeful attacker. We propose instead to identify the strategic goals an attacker might have, and to choose the

system inputs accordingly. In Table VI we show a first draft of such goals. We have separated out two dimensions: the type of damage an attacker intends to cause, and the attack mode, e.g., whether the attack shall remain stealthy or not. Note that, in particular for stealthy attacks, the input patterns might not be obvious. Then simulation also has an important role to play to find and optimize the system parameters accordingly.

It is a joint task for safety engineers and security engineers in cooperation with members of agencies such as the BSI (Bundesamt für Sicherheit in der Informationstechnik), the relevant authority in Germany, to assess the likelihood of such attacks: the first group can assess the necessary resources (e.g., knowledge, access to equipment) for an attack category, while the latter can assess whether corresponding groups with the respective strategic goals are able to obtain these resources.

C. Exploration and Assessment of Mitigation Measures

In Section IV we have seen how measures that act on the physical part of the service can play an important role to mitigate the impact when key establishment fails.

Simulation can either be part of the measure itself as with cyber-physical attack detection in form of a digital twin or it can play a crucial role to validate the measure. There are several facets here: first, to validate whether the physics behind the method will indeed work. Second, to simulate and validate the actions of ground personnel in case of an alarm, e.g., to estimate the time it takes for them to deactive the respective machine. And third, to validate whether the time between the alarm and the deactivation is sufficiently short to reduce risk before damage is caused. Finally, co-simulation can be used for an overall validation. Again, simulation can also be used for parameter optimization. For any attack detection system it will be important to consider the evaluation criteria considered in [14]: the trade-off between the maximum deviation of critical system variables per time unit imposed by undetected attacks, and the expected time between false alarms.

VI. Related Work

Safety and Security Process: In view of the increased use of wireless communication in transportation there has been a long-term undertaking to integrate both safety and security into norms and standards, and to devise appropriate methods for threat analysis and risk assessment. Important methods, originally from the automotive domain, are the one of the project EVITA [15], the one of the project HEAVENS [16], and the SAHARA method [17], which combines HARA (hazard analysis and risk assessment) from the safety domain with STRIDE [18] from the security domain. The standardization efforts in the automotive domain have culminated in the recent ISO/SAE 21434 Automotive Cybersecurity Standard (c.f. [19]). This has led to a maturing of the methods into tool-chains [20]. However, these methods and tools focus on high-level system aspects, and do not adequately capture and structure the level of key establishment.

Authenticated Key Establishment Protocols: The Diffie-Hellman key exchange [21] is the first key establishment protocol in the public key setting; indeed it was put forward at the same time as the idea of public key cryptography itself. Since then much effort has been gone into how to design and verify *authenticated key establishment protocols*, which can be used over an open channel and in the presence of an active adversary to securely establish a key [9], [11], [12]. It became clear that such protocols are vulnerable to subtle attacks, and it is now standard to formally verify security protocols by state-of-the-art symbolic protocol tools such as Tamarin [7], [22] or ProVerif [23] and/or to prove them secure in a cryptographic security model [10].

Both, design and verification of key establishment protocols is an ongoing activity: not least since the advent of quantum computing will also affect the protocols in use today. For example, the novel PQXDH (Post-Quantum Extended Diffie-Hellmann) protocol provides post-quantum forward secrecy while still being based on the discrete logarithm problem [24].

While there is very rigorous methodology to design and verify security protocols, most of the activities are focussed on the protocols themselves under standard assumptions (e.g., concerning key reveals) motivated by cyber-only applications. Here we have focussed on outside the box challenges unique to our M2M setting and the potential of cyber-physical measures.

Secure Device Pairing: SDP has been an active research field ever since it was put forward by Stajano and Anderson in 1999 (c.f. [25]). Moreover, SDP schemes have been widely adopted for IoT and personal devices. Mirzadeh et al. [26] review device pairing protocols and their security, including group device pairing. Fomichev et al. [6] provide terminology and foundations for a classification and comparison of existing SDP schemes, and a comprehensive survey thereof. SDP has been less investigated and put to practice in the context of pairing up large machines of different security domains. We have used here SDP in a hybrid form, employing an AKE protocol. The TAGA channel provides a new type of OoB channel, which in our context turned out to be insufficiently secure but which provides a second line of defence.

VII. Conclusions and Future Work

In this work we have addressed the gap of how to engineer and validate key establishment methods in safety-critical M2M settings. We have put forward to work with connection compromise states, which define how key establishment can fail and allow for a more fine-grained integration with cyber-physical mitigation measures. We have also seen that in our setting there is a range of measures available that work well, in particular when there are complex trust assumptions due to participants coming from different security realms. Our examples have shown that the consequences of threats against key establishment such as LTKCs can be subtle and unexpectedly large in M2M settings.

Concerning security verification the protocols in use should undergo the rigorous verification process of the cryptographic protocol community. The attacker model can be adapted so that

it suits the channel used, which in turn has to be rigorously investigated. Here we have reverted to the standard Dolev-Yao model, who has full control of the network. However, in a parallel work we investigate how this can be slightly relaxed for the TAGA channel. We also explore a solution, which will be fully local and not depend on global key management [8]. We will also explore post-quantum security for our setting. Another important point for future work is to find methods that allow us to assess the likelihood of LTKCs and other threats against key establishment in a systematic way.

Moreover, this paper has demonstrated that simulation plays an important role in the process to develop and validate a key establishment method for security and safety. Of course, the activities described here can be followed by bench/live tests, and formal verification where necessary. In particular, we wish to investigate whether and how statistical model-checking [27] can be made use of in the tool-chain: to be able to verify integrated safety and security properties such as: "Safety mitigation kicks in before attack causes harm with probability $>$ P".

References

[1] S. Fröschle and M. Kubisch, "Security process for adopting machine to machine communication for maintenance in transportation with a focus on key establishment," in *Proceedings of SIMUL 2023: The Fifteenth International Conference on Advances in System Simulation*. Thinkmind Digital Library, 2023, pp. 50 – 58.

[2] S. B. Fröschle, M. Kubisch, and M. Gräfing, "Security analysis and design for TAGA: a touch and go assistant in the aerospace domain," *CoRR*, vol. abs/2004.02516, 2020. [Online]. Available: https://arxiv.org/abs/2004.02516

[3] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 3rd ed. CRC Press, 2021.

[4] G. Lowe, "A hierarchy of authentication specification," in *10th Computer Security Foundations Workshop (CSFW '97), June 10-12, 1997*. IEEE Computer Society, 1997, pp. 31–44.

[5] C. Boyd, A. Mathuria, and D. Stebila, *Protocols for Authentication and Key Establishment*, 2nd ed. Springer Publishing Company, 2020.

[6] M. Fomichev, F. Álvarez, D. Steinmetzer, P. Gardner-Stephen, and M. Hollick, "Survey and systematization of secure device pairing," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 517–550, 2018.

[7] B. Schmidt, S. Meier, C. Cremers, and D. Basin, "Automated analysis of Diffie-Hellman protocols and advanced security properties," in *25th IEEE Computer Security Foundations Symposium, CSF 2012*. IEEE, 2012, pp. 78–94.

[8] S. Fröschle and M. Kubisch, "Three taps for secure machine-to-machine communication: Towards high assurance yet fully local machine pairing," in *Proceedings of the Sixth Workshop on CPS&IoT Security and Privacy*, ser. CPSIoTSec'24. New York, NY, USA: Association for Computing Machinery, 2024, p. 125–133. [Online]. Available: https://doi.org/10.1145/3690134.3694824

[9] A. P. Sarr, P. Elbaz-Vincent, and J.-C. Bajard, "A secure and efficient authenticated Diffie–Hellman protocol," in *Public Key Infrastructures, Services and Applications*. Springer, 2010, pp. 83–98.

[10] A. P. Sarr and P. Elbaz-Vincent, "On the security of the (F)HMQV protocol," in *Progress in Cryptology – AFRICACRYPT 2016*. Springer International Publishing, 2016, pp. 207–224.

[11] S. Blake-Wilson, D. Johnson, and A. Menezes, "Key agreement protocols and their security analysis," in *Crytography and Coding*. Springer, 1997, pp. 30–45.

[12] S. Blake-Wilson and A. Menezes, "Authenticated Diffie-Hellman key agreement protocols," in *Proceedings of the Selected Areas in Cryptography*, ser. SAC '98. Springer-Verlag, 1999, pp. 339–361.

[13] J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, N. O. Tippenhauer, H. Sandberg, and R. Candell, "A survey of physics-based attack detection in cyber-physical systems," *ACM Comput. Surv.*, vol. 51, no. 4, jul 2018. [Online]. Available: https://doi.org/10.1145/3203245

[14] D. I. Urbina, J. A. Giraldo, A. A. Cardenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg, "Limiting the impact of stealthy attacks on industrial control systems," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 1092–1105. [Online]. Available: https://doi.org/10.1145/2976749.2978388

[15] R. A., D. Ward, W. B., I. S., R. Y., M. Friedewald, T. Leimbach, A. Fuchs, S. Gürgens, O. Henniger, R. Rieke, M. Ritscher, J. Broberg, L. Apvrille, R. Pacalet, and G. Pedroza, "Security requirements for automotive on-board networks based on dark-side scenarios," 2009, EVITA Project, Deliverable D2.3, v.1.1.

[16] M. M. Islam, A. Lautenbach, C. Sandberg, and T. Olovsson, "A risk assessment framework for automotive embedded systems," in *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*. ACM, 2016, pp. 3–14.

[17] G. Macher, H. Sporer, R. Berlach, E. Armengaud, and C. Kreiner, "Sahara: A security-aware hazard and risk analysis method," in *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2015, pp. 621–624.

[18] A. Shostack, "Experiences threat modeling at microsoft," in *Workshop on Modeling Security*, Toulouse, September 2008.

[19] G. Macher, C. Schmittner, O. Veledar, and E. Brenner, "ISO/SAE DIS 21434 automotive cybersecurity standard - in a nutshell," in *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops*, A. Casimiro, F. Ortmeier, E. Schoitsch, F. Bitsch, and P. Ferreira, Eds. Cham: Springer International Publishing, 2020, pp. 123–135.

[20] C. Schmittner, B. Schrammel, and S. König, "Asset driven ISO/SAE 21434 compliant automotive cybersecurity analysis with threatget," in *Systems, Software and Services Process Improvement*, M. Yilmaz, P. Clarke, R. Messnarz, and M. Reiner, Eds. Cham: Springer International Publishing, 2021, pp. 548–563.

[21] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[22] D. Basin, C. Cremers, J. Dreier, and R. Sasse, "Tamarin: verification of large-scale, real world, cryptographic protocols," *IEEE Security and Privacy Magazine*, 2022. [Online]. Available: https://hal.science/hal-03586826

[23] B. Blanchet, "Modeling and verifying security protocols with the applied pi calculus and ProVerif," *Foundations and Trends in Privacy and Security*, vol. 1, no. 1–2, pp. 1–135, Oct. 2016.

[24] "The PQXDH key agreement protocol, revision 3," 2024, called 31/03/2024. [Online]. Available: https://signal.org/docs/specifications/pqxdh/

[25] M. Li, W. Lou, and K. Ren, "Secure device pairing," in *Encyclopedia of Cryptography and Security*. Springer US, 2011, pp. 1111–1115.

[26] S. Mirzadeh, H. Cruickshank, and R. Tafazolli, "Secure device pairing: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 17–40, 2014.

[27] E. M. Clarke and P. Zuliani, "Statistical model checking for cyber-physical systems," in *Automated Technology for Verification and Analysis*, T. Bultan and P.-A. Hsiung, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1–12.