Security Process for Adopting Machine to Machine Communication for Maintenance in Transportation with a Focus on Key Establishment

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 illustrate our approach 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, simulation, threat and risk analysis, 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. It is clear that 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. The state of the art of key establishment offers two approaches: either we can make use of an *authenticated key establishment protocol* [1] but have to deal with the challenge of how to securely set up and manage a pre-shared security context. Or, we can try to translate the idea of *secure device pairing* [2]–[4] into our context of safety-critical machines.

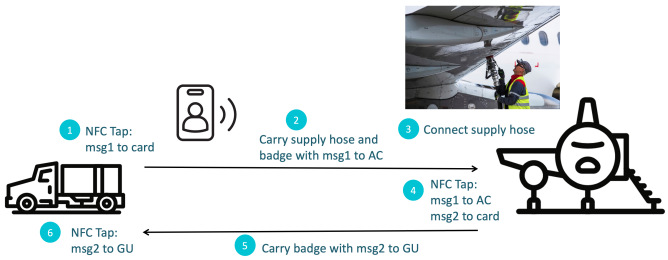


Fig. 1. Pairing up a ground unit and an airplane

To adopt M2M communication for ground processes at airports a *touch and go assistant in the aerospace domain (TAGA)* is currently under development. Each aircraft and ground unit is equipped with a TAGA controller that contains a secure element for cryptographic operations and a Near Field Communication (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 Fig. 1.

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-

Copyright (c) IARIA, 2023. ISBN: 978-1-68558-102-2

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). 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: To sum up, while there is a vast design space of key establishment methods 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.

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 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 V we draw conclusions and discuss future work.

II. Key Establishment for Vehicle to Service Unit Communication

Setting: We first define the problem setting. As shown by example in Fig. 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.

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.

We assume that the communication is conducted over a wireless channel (such as Wi-Fi IEEE 802.11), and that a corresponding protocol to ensure data confidentiality and integrity during data transmission is already determined (such as AES-GCM for Wi-Fi IEEE 802.11). 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 [5]. 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. 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* [1]; the second is to make use of a *Secure Device Pairing (SDP) scheme* [4]. As we will see later a combination is also possible.

AKE protocols [1] are by now well-investigated, and there exist many standardized protocols that come with formal 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

Copyright (c) IARIA, 2023. ISBN: 978-1-68558-102-2

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 [4] schemes make do without a pre-shared security context but instead rely on so-called Out-of-Band (OoB) channels to safeguard against man-in-the-middle 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: 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.

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* are described in Table II and illustrated in Fig. 2. 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.

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

1)	*Man-in-the-middle (MitM).* 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.* 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:* A peer P shares a connection secured by 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.

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 IV since this is where simulation plays a crucial role throughout. Part (1) will only 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 [6].

III. TAGA: A TOUCH AND GO ASSISTANT IN THE AEROSPACE DOMAIN

A. 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 Fig. 3 for the case when the Diffie-Hellman (DH) 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 he will also carry a supply hose; e.g. for pre-conditioning he will carry the air supply hose.

Copyright (c) IARIA, 2023. ISBN: 978-1-68558-102-2

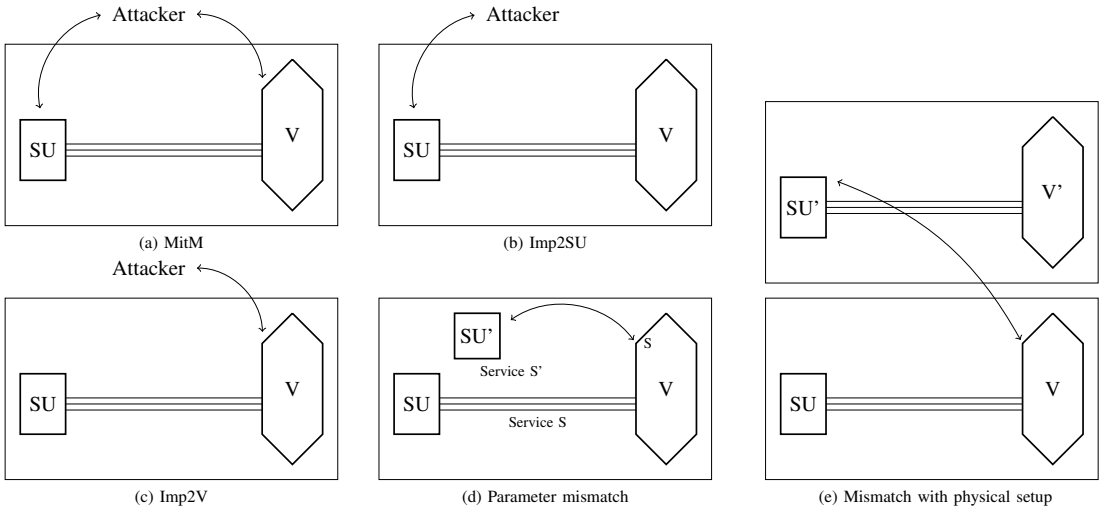
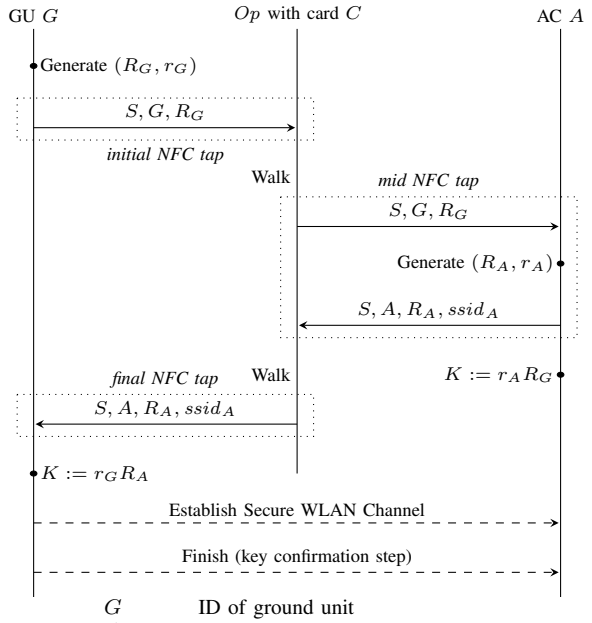


Fig. 2. Illustration of the connection compromise states



G	ID of ground unit
A	ID of aircraft
S	service name of ground unit
$ssid_A$	SSID of the aircraft's WLAN
R_G, r_G	public and private DH key of G
R_A, r_A	public and private DH key of A
K	resulting session key

Fig. 3. TAGA pairing with Diffie-Hellman key exchange

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 activates the ground unit; e.g. for pre-conditioning he switches 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 authenticity of 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 man-in-the-middle attack against the basic Diffie-Hellman exchange over the TAGA channel.

Example 1 (MitM 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 his 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 Fig. 4. 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.

Copyright (c) IARIA, 2023. ISBN: 978-1-68558-102-2

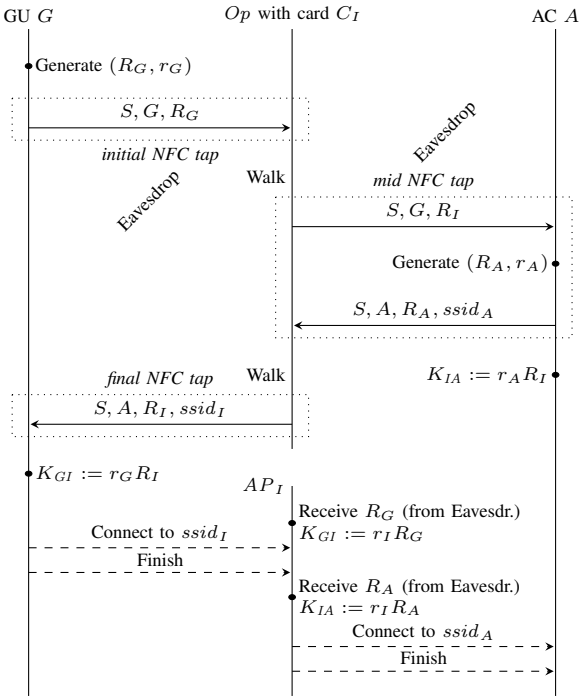


Fig. 4. Man-in-the-middle attack by card swapping and eavesdropping

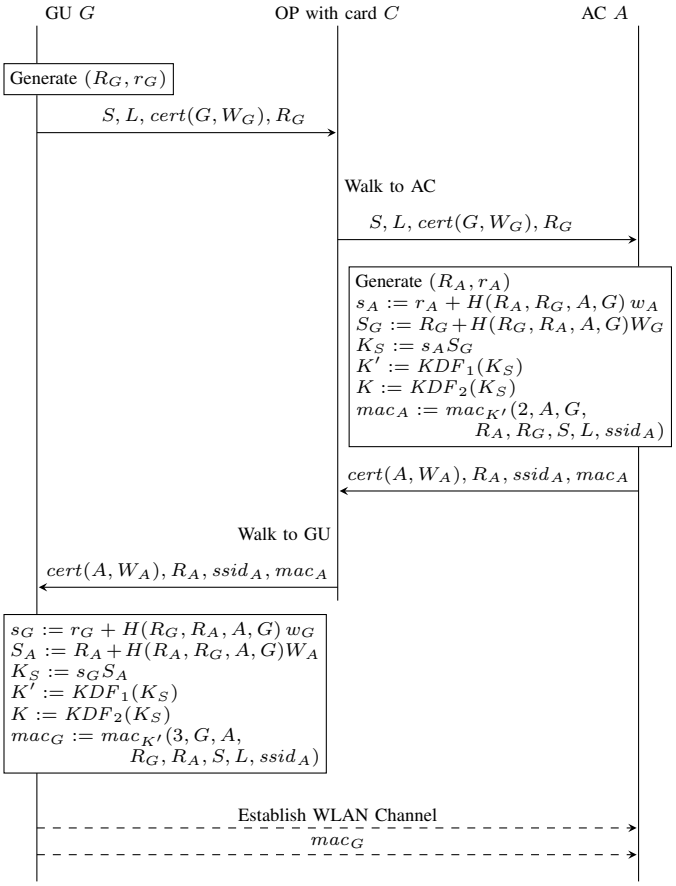


Fig. 5. TAGA pairing based on the FHMQV protocol

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

Estimating the Safety Impact: To estimate the severity of impact of a MitM 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 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: he 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.

B. 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.

Fig. 5 shows TAGA based on the *Fully Hashed Menezes-Qu-Vanstone protocol (FHMQV)* [7], [8]. 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 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 he has 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 his own counterfeit card C_I for the card of G's operator. Moreover, the attacker sets up NFC eavesdropping capability, and his own WLAN access point AP_I within range of L. 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

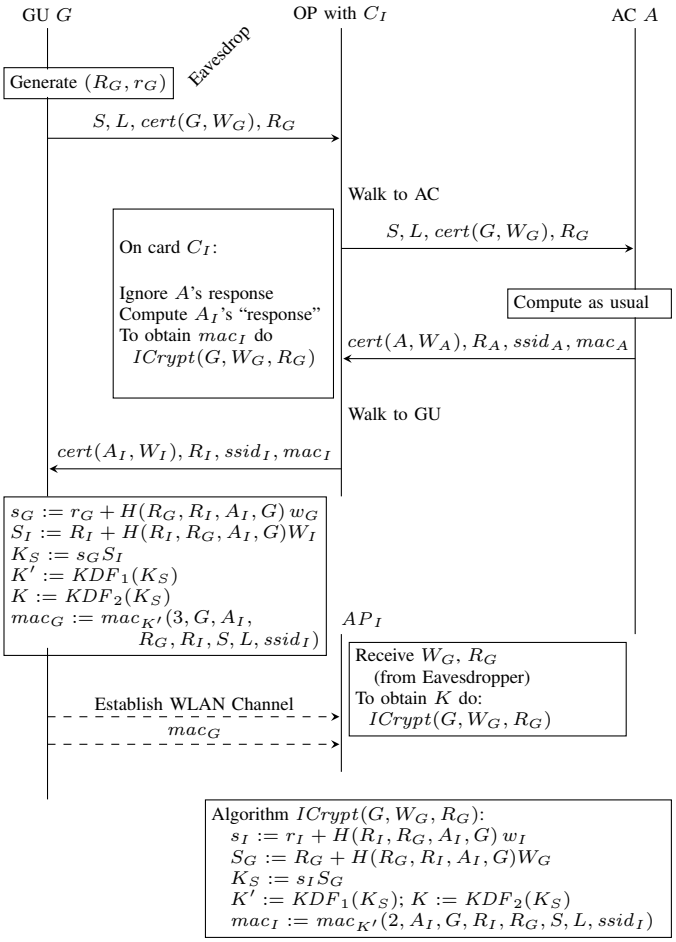


Fig. 6. Impersonation to ground unit with LTKC of any aircraft

(r_I, R_I), and the SSID $ssid_I$ of the attacker's WLAN.

Then the attacker can proceed as shown in Fig. 6: he simply establishes 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 a Imp2SU connection compromise, and one may hope that this comes with less safety impact than MitM. However, Imp2SU still allows the attacker to feed any sensor values he likes 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

Copyright (c) IARIA, 2023. ISBN: 978-1-68558-102-2

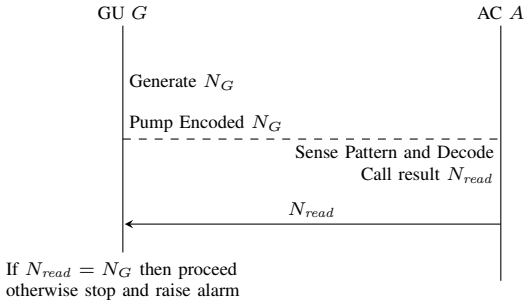


Fig. 7. Physical Challenge/Cyber Response

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 MitM attack, the attacker is not able to obtain sensor values sent by the aircraft, and, hence, he is not able to optimize the attack based on such information.

Given the potential safety impact and scale of the attack (given one LTKC) 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.

C. Refinement: Including Mitigation Measures

We explore several ways of how to implement detection against Imp2SU (in the absence of MitM). The following measure 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.

Example 6 (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 Fig. 7. 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 by the 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.

The space of nonces must be sufficiently large to reduce the risk of guessing attacks: even when the attacker cannot receive the physical signal he can always guess the nonce N_G and send it back via a cyber channel he has 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.

IV. 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.

1) 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).

2) 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.

Copyright (c) IARIA, 2023. ISBN: 978-1-68558-102-2

TABLE III
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).

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 III 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.

3) Exploration and Assessment of Mitigation Measures: In Section III-C we have seen by example that measures that act on the physical part of the service can play an important role to mitigate the impact when key establishment fails. The following measures might be options to protect against Imp2SU or even MitM:

- *Physical Challenge/Cyber Response (only against Imp2SU):* as described in Example 6.
- *Time-based Detection (only against Imp2SU):* Due to the interweaving with the physical setup an attacker who carries out an Imp2SU attack will typically need to initiate a fake key establishment session with the vehicle (c.f. Example 4) as part of the attack. This session will never be completed, and hence, the vehicle could raise an alarm when a session is still pending after an unusally long time. Operators could then check what is going on, and, e.g. deactivate the service unit before damage occurs.
- *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.
- *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 [9], [10]. 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.

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 [10]: the trade-off between the maximum deviation of

Copyright (c) IARIA, 2023. ISBN: 978-1-68558-102-2

critical system variables per time unit imposed by undetected attacks, and the expected time between false alarms.

V. Conclusions and Future Work

We hope this paper has demonstrated that a key establishment method can be systematically developed and validated for security and safety, and that simulation plays an important role in this process. Of course, the activities described here can be followed by bench/live tests, and formal verification where necessary. In particular, we will investigate whether and how statistical model-checking [11] can be made use of in the tool-chain: to be able to verify integrated safetey and security properties such as: "Safety mitigation kicks in before attack causes harm with probability $>$ P".

References

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

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

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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

[10] 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, p. 1092–1105. [Online]. Available: https://doi.org/10.1145/2976749.2978388

[11] 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.

Copyright (c) IARIA, 2023. ISBN: 978-1-68558-102-2