Secure data recording and bio-inspired functional integrity for intelligent robots

Sebastian Taurer[1], Bernhard Dieber[1], Peter Schartner[2]

***Abstract*— As modern robots become more intelligent, also their use will broaden in public and professional areas. While the aim is to make robots beneficial to humans and society, using those complex machines in complex environments will eventually lead to incidents. To enable forensic investigations, ethical evaluations and transparent function of intelligent robots in a society, we contribute the concept of a secure robot data recorder that is similar to a flight data recorder in airplanes. However, since robots work in a highly networked and uncontrolled environment, our concept pays special attention to security and tamper proofness. In addition, we extend the concept with an approach inspired by cockroaches to increase the functional integrity of the robot. We present a prototype implementation along with discussions on the required properties and limits of secure data recording.**

I. Introduction

As robots gain more autonomy, accountability and integrity measures become more and more important. This is true in more than one perspective. First and most obvious, there is the necessity to record the state and actions of a robot leading up to an incident. So if — despite all safety measures — a harmful incident with a robot happens, it is beneficial (and likely required in future) to have data to run forensic investigations on. Second, from an ethical perspective, data recording on the decisions and actions of a robot can help in understanding its motives and find improvements to its morally or socially relevant behavior [1]. Third, intelligent robots need to be transparent [2], [3] in the sense, that they must be explainable for humans. A data recorder can support this by providing the required data and indications on the class of data (e.g., if privacy-related data is collected).

Recently, the European Parliament has issued a resolution where the topics of ethical, societal and legal issues in the use of robotics and autonomous systems are addressed [4] including issues of accountability and liability where data recording can be seen as a crucial contribution also with the proposed measures for mandatory robot insurances.

In the context of intelligent robots, it has also to be considered, that those will be increasingly targets of cyber-attacks and tampering. Thus, data recording has to be secured against such manipulations down to the level of the recorded data itself.

We propose an architecture of a secure data recording mechanism and device for intelligent robots. It connects to the main robot controller and records data relevant to recreate the situations the robot was in. It stores this data securely on internal memory using cryptographic operations to ensure confidentiality, integrity and authenticity of the data. Further, we argue that such a device can additionally be used to act as an external integrity and safety monitoring device, which can detect certain unintended states in the robot and perform a limited set of countermeasures against that. Here, we adopt a bionic approach inspired by cockroaches. Finally, we describe an initial prototype on real hardware and discuss the current possibilities and limits of our implementation.

In the rest of this paper, we survey data recorders in section II, present our approach in section III, describe the bionic amendment in section IV and our prototype in section V. A discussion of our concepts is presented in section VI.

II. Related Work

Data recorders (also called black boxes) have been used in aviation from the end of the 50s [5], the first being constructed by David Warren [6]. It records data for use in case of incidents for forensic investigations. Traditionally, telemetry and control data of the airplane but also cockpit communication and sometimes also video data is logged.

Flight data recorders are typically put in highly robust housings [7]. This protects the sensitive recordings from impact forces, water, excessive heat and flames. Also, special protection for the memory itself has been developed [8].

The concept of data recorders has also been transferred successfully to other areas like railway operation [9]. For cars, special solutions to log state and steering commands have been proposed [10], [11].

In digital systems, data recording has been proposed for embedded systems[12], but also for pure software [13].

III. Approach

We now describe our secure data recorder starting with the basic requirements and continuing with the chosen design.

A. Requirements for a robotic data recorder

A secure data recorder for autonomous robots (traditionally, this is also referred to as a black box) needs to fullfil several requirements in order to be useful. Note, that we do not focus on which data exactly is recorded since this is a separate topic of system diagnosis.

Requirement R1: Availability. A very obvious requirement is, that the data recording must be done in a reliable way meaning that no data is lost or corrupted and all the data must be available when it is needed.

[1]JOANNEUM RESEARCH - Institute for Robotics and Mechatronics - Robotic Systems - Klagenfurt, Austria, firstname.lastname@joanneum.at

[2]Alpen-Adria Universität Klagenfurt - Institute of Applied Informatics

Requirement R2: Data confidentiality, integrity and authenticity. The data stored in the black box should be kept safe from intentional and unintended modification. In contrast to airplanes, an autonomous robot has bigger attack surfaces because it is highly connected and exposed to a rather uncontrolled environment. Thus, we argue that a state recording mechanism for robots — besides recording data — must also keep this data secure. Specifically, the recording should ensure confidentiality, integrity and authenticity of the stored data. Confidentiality is necessary since some data parts may contain sensitive private or business information depending on the application environment of the robot (part of the state information of a medical care robot could also include the patient it is working with). Data integrity must be ensured for forensic purposes since it is important that data has not been damaged or modified after recording in order to be forensically valuable. Authenticity of the data is required to make sure that the data used in forensic proceedings has actually been produced by the robot under investigation.

Requirement R3: Physical requirements and tamper resistance. Besides the digital safekeeping of the collected data, also physical access to the device and modifications of the parametrization should be restricted. This includes the physical destruction or damaging the device but also the unauthorized removal. The black box must be secured against physical damage and destruction as consequences of accidents. This however, is done in mechanical and safety engineering and will not be closer examined in this work. Further, the access to the internal configuration (including also cryptographic keys, certificates, ...) must follow defined workflows. In addition, the implementation and the underlying platform needs to expose as low security risks as possible.

B. Approach overview

To fulfill the requirements stated above, we envision our data recorder as a separate computing unit outside of the robot controller to which it connects via a wired communication channel. The controller reports state data to the data recorder where it is securely stored.

We have developed a cryptographic scheme and workflows to interact with the data recorder in a controlled way. This includes how it is (de-)commissioned and how cryptographic keys and certificates are generated and managed. The storage of data follows a specified scheme, which ensures data integrity, confidentiality and authenticity and uses state-of-the-art cryptographic techniques to achieve this.

Since the data recorder receives a constant stream of data, which describes the state of the main controller, it can also be used as a monitoring unit for the controller ensuring basic functional integrity properties. This is explained in more detail in section IV.

C. Setup

Figure 1 shows the conceptual overview of the overall system. It consists of the controller (which can be standard computing hardware) and the robot itself (under which we generalize all types of robot hardware like serial manipulators or mobile platforms). Connected to both is the Black Box (BB), i.e. the robot data recorder. The BB receives a constant stream of state data from the controller. The link is assumed to be a wired connection like e.g. USB, other serial communication or Ethernet (for security reasons preferably separated from other networks). The reason for the connection between the BB and the robot is explained in section IV. The BB itself is a piece of computing hardware, which – from a security point of view – is hardened against tampering and cyber attacks. Additionally, no remote connection to the BB is possible for the same reason. The direct connections within the robot system make it possible that a disconnection is detected by the participants and actions can be triggered. Transmitting the state data at least once in a predefined time interval ensures the detection of a broken or disconnected communication link. The communicated messages are secured and kept confidential with the help of Diffie-Hellman key exchange protocol to securely exchange cryptographic keys which are used to encrypt/decrypt the communication data. If the robot system (or at least the data recorder hardware) is surrounded with a tamper safe housing, the BB can be connected to that housing to notice malicious physical intrusions.

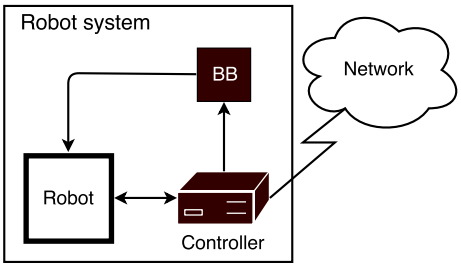

Fig. 1. The robot system and its components with a connection to the outside world.

D. Initialization

The robot system must pass through an initialization phase to receive configuration information. All generated data is unique for each single robot system. This includes cryptographic keys and certificates to guarantee features like confidentiality, integrity and authenticity. The steps of this phase must be done in a secure environment to avoid injection of malicious information or eavesdropping.

1) Certificates: A digital certificate binds the ownership of a public key to an identity. A certificate authority (CA) is a point of trust and an entity that issues digital certificates. The organisation, which runs the BB, needs to have a digital certificate in order to get access the the BB and its log-file. Additionally, the CA has to create a digital certificate for every authorised operator (i.e., a person, which maintains the robot). The private key of the certified public key is stored on a smartcard for secure usage. To control the permissions and privileges of every authenticated operator, attribute certificates (AC)[1] are used. An AC may contain attributes that

[1]Defined in RFC 5755

specify group membership, role, security clearance, or other authorization information. The following certificates must be transferred to the BB during the initialization phase:

- Certificate of the certificate authority (CA)
- Certificate of the organisation, which runs the BB
- Certificate(s) of the authorised operator(s)

The certificate of the CA is needed to verify all the other certificates, which are issued by the CA. So, the key usage of this certificate is signature validation, but the purpose of the organisation's certificate is providing a key for encipherment. The certificates of the authorised operators are needed to run an authentication protocol in maintenance situations.

2) Key generation: Authenticity can only be ensured with the help of digital signatures. Therefore, the BB has to generate a key-pair (pk, sk) for creating and verifying digital signatures. The public key pk, which is used to verify the digital signature, is bound to the BB by merging it with the unique identifier of the BB and other informations in a digital certificate. This certificate is signed by the CA and transmitted to the organisation at the end of the initialization phase. The secret key sk is used to create digital signatures to authenticate the log-file entries. It is important that the secret key sk is generated and stored inside the BB and never leaves it.

E. Operation

During operation, the robot performs its tasks, the controller is controlling the robot and sending the system state to the BB. The BB is logging and monitoring the received system states to generate a reproducible state history. Additionally cryptographic operations like encryptions, signature creations and hash value calculations are performed.

Hereafter, we write $E(m, k)$ for (symmetric or asymmetric) encryption of the message m with a given key k. In case of asymmetric cryptography, we write pk or sk for the public or private key of an entity. For digital signatures, let $S(m, sk)$ be the signature function where the message m is signed with the help of the private key sk to output the signature s. A signature s can be verified with the function $V(m, s, pk)$ that needs the public key as additional input to the signature to output either $true$ or $false$, depending on whether or not the signature was cryptographically valid. The symbol $x \parallel y$ means the concatenation of the data items x and y in a way that x and y can both be recovered uniquely from the compound representation $x \parallel y$.

In the following we let our description be abstract, yet emphasize that possible cryptographic schemes are AES [14] for symmetric encryption, RSA [15] for asymmetric encryption or signature creation and SHA-256 [16] for hashing.

The system state of the robot system is a collection of information where each value represents one single component of the system state. The controller knows, which information is part of the system state, and generates it by reading the corresponding values. The state data is sent from the controller to the BB at least once in a predefined time interval - like a heartbeat. The BB also reads information from the tamper detection sensors mounted on the body housing, if existing. All data, which is received by the BB, is provided with a time-stamp and summarized as received data rd_{i}.

The content of the log-file is shown in fig. 2. The first entry in the log-file is the public key of the BB pk_{BB} to verify digital signatures created with the secret key sk_{BB}. The following entries have a common structure: 1) the encrypted data that either represents the received data rd_i or an block specific encryption key k_j, 2) the hash-value based on the concatenation of the encrypted data with the previous hash-value to build a hash-chain, 3) the digital signature to authenticate the encrypted data. Several entries are combined in a block to repeatedly change the encryption key for the encryption of the received data. This step increases the security of the log-file data because not every entry is encrypted with the same key. Every block b_j starts with an header entry which contains the block specific encryption key k_j to encrypt the received data inside the block. This key is stored in the block header asymmetrically encrypted with the public key of the organisation's certificate created in section III-D.1. So, only the organisation can decrypt the block specific encryption keys with its private key. The received data is encrypted with an symmetric encryption algorithm for better performance. The hash-value hd_i (resp. hk_j) prevents manipulation of the data. The hash-chain establishes a link between two consecutive log-file entries, which guarantees that an entry e_i with $i < j$ is the predecessor of an entry e_j and prevents the insertion of new block entries between existing ones. The authenticity of the log-file data can only be established with digital signatures. In no other cryptographic primitive (e.g. Message Authentication Code (MAC)) is a link between the output (ciphertext or signature) and its originator. The digital signature provides an evidence that the content of the log-file was created by this specific BB, because the generated private key of section III-D.2 is exclusively used and owned by the BB. The input of the signing function S for signature creation of the log-file data is the beforehand calculated hash-value.

The encryption of the received data $ed_i := E(rd_i, k_j)$, the hash-value calculation and hash-chain generation $hd_i := H(ed_i \parallel hd_{i-1})$ and the digital signatures $sd_i := S(hd_i, sk_{BB})$ guarantee confidentiality, integrity and authenticity of the data which is important for section III-G.

F. Maintenance mode

The maintenance mode must be activated when an operator wants to perform maintenance actions, updates or changes on the robot system and its components. Maintenance on the robot itself could be the replacement of some broken parts, maintenance on the controller could be a software update and maintenance on the BB could be the transmission of a new operator certificate. But in every case, the BB must be informed about the ongoing work.

1) Enable maintenance mode: To enable the maintenance mode, an authorised operator authenticates towards the robot system. This is done using a smartcard and a smartcard reader. The private key on the smartcard relates to the public

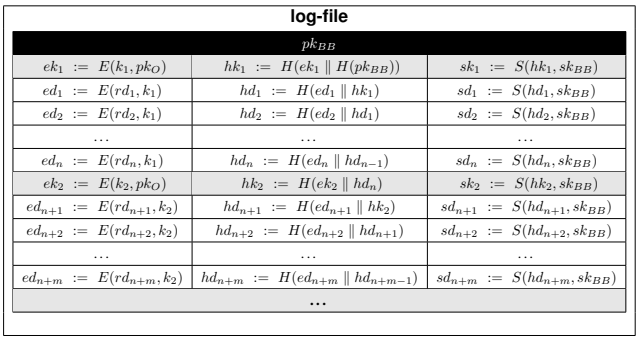

Fig. 2. The logfile consists of the encrypted data, a hash-chain of the encrypted data and digital signatures. In case of a forensic investigation authenticity and integrity of the encrypted data can be verified.

key of the operator's certificate, which is already known to the BB (see Section III-D). After connecting the smartcard reader to the BB, the operator is able to authenticate towards the BB. The smartcard sends its identity to the BB, which searches in its database of operator-certificates for the matching identity. Iff the certificate is found in the list, the authentication process starts with a challenge-response-protocol. After the successful termination of the protocol, the BB is in maintenance mode and the operator gets access.

2) Certificate revocation: A certificate is valid if its digital signature can be correctly verified and it is not expired. In general, servers accept requests of clients with valid certificates if the certificate is not listed in the certificate revocation list (CRL). The BB only accepts requests when the certificate is valid and additionally stored in the list of trusted certificates transmitted in section III-D.1. Because there is no remote connection to the BB, a CRL can not be updated and therefore not be used in our prototype. Thus, changes in the list of trusted certificates must be realized by manually adding, removing or replacing a certificate on the BB.

G. Forensic investigation

A forensic investigation is done in case of an accident or unexpected behaviour of the robot system. An investigator analyses the factors and tries to reconstruct the course of events, which led to the unwanted situation. Therefore, an authorised operator has to read out the log-file from the BB to provide it to the investigator. Before studying the log-file data, the investigator can perform the following security checks. The certificate of the BB, created in section III-D.2, contains the public key to verify the digital signatures of the log-file data. This public key must be the same public key which is written in the log-file. If the keys are different the log-file originates form another BB. With the help of the public key, the investigator is able to verify the digital signatures in the log-file to check if all the entries were recorded by this particular BB. After that, the hash-chain can be checked by reconstructing and comparing it with the hash-chain in the log-file. The reconstruction is done by calculating a hash-value for every entry as shown in figure 2. If the reconstructed hash-chain is the same as the one in the log-file the investigator can be sure that no error or manipulation of the log-file data has been occurred. Because the data in the log-file is encrypted, the organisation has to decrypt the block headers with its private key to extract the block specific keys. With the help of the block specific keys, the investigator is able to decrypt the entries of each block by using the correct key.

After successfully verified digital signatures, checked hash-values of the hash-chain and decrypted log-file data, the investigator can start analysing the recorded actions and state data to reconstruct the course of events.

Data loss scenarios: If data is lost due to fire or other damage, only a partial reconstruction of events is possible. If a block entry is damaged, it is not possible to verify the hash-value of the following block entry. However, all further entries and their sequence can be verified. The amount of data, which's integrity can be verified in such a case depends on how often the digital signature has been applied. In case that each block entry was signed separately, a full verification of all non-damaged entries is possible. If the whole block was signed (or its hash, respectively), one damaged entry is enough to prevent a signature verification. This is rooted in the nature of digital signatures. Thus, finding the trade-off between performance and security is a crucial decision.

One possible addition to reduce data loss would be to add an error-correcting code (ECC) [17]. This however, would also increase the amount of data, which needs to be stored but in some cases all of the errors can be corrected. Suppose the operation mode of the symmetric encryption/decryption algorithm is the Cipher Block Chaining (CBC) mode. One of the characteristic of this mode is that after decryption a single bit error in the ciphertext-block c_i has influenced the whole plaintext-block p_i and additionally the next plaintext block p_{i+1}. So, the plaintext p_i can not be reconstructed and the next plaintext block p_{i+1} has a single bit error on the same position as the error prone ciphertext c_i. Correcting the bit error in the plaintext p_{i+1} with an ECC would make it possible to correct also the bit error in the ciphertext c_i which makes it possible to successfully decrypt c_i to recover p_i. If it is not possible to correct all the bit errors in the plaintext p_{i+1} and therefore not in c_i an ECC applied on the ciphertext c_i could correct the remaining bit errors.

IV. Bio-inspired approach to functional integrity

So far, we have described our BB concept as a pure data recording device. However, since it has access to the system state in near-realtime, we envision a second functionality, which ensures functional integrity of the overall system.

We have taken inspiration from common cockroaches. As it is well-known, this species is exceptionally robust and hard to kill. On closer study, this is not only rooted in the hard outer shell of those insects but also in the construction of their nervous systems [18]. Instead of a single brain, the cockroach has a second, smaller ganglion (an accumulation of nerve cells) close to its rear legs. Its function is to control the flight behavior of the insect. It is either triggered by

impulses from sensory hair or by the main brain. Here, the larger ganglion acts as an inhibitor, so as long as it is functional, it will prevent the rear ganglion from initiating an escape. However, as soon as the main brain is damaged, the inhibition is no longer present and the rear ganglion will initiate the flight behavior.

We transfer this concept to our BB, which in this case takes the role of the rear ganglion. But instead of only a single emergency behavior, we use a set of rules to trigger a corresponding action if the received system state appears abnormal. Actions can range from cutting power to the motors (or an emergency stop of the robot, respectively) to interrupting the outside network connection if a cyber attack or tampering attempt is suspected.

In addition, we also transfer the inhibition functionality of the main brain (the controller) towards the BB. If for any reason, the controller stops transmitting state data (or if the frequency drops significantly), the BB can also trigger a specific action. The same is true if the device detects tampering (enabled by special tamper-pins in hardware).

Note, that the storage and functional integrity must be implemented in two separate subsystems although they make use of the same data stream. The two components are subject to different safety integrity levels and must be able to function independently. Thus, they may make use of the same data connection but may not interact or influence each other.

V. Prototypical implementation

In this section, we describe a prototypical implementation of our approach. Note, that we do not focus on which data is recorded since this is very application-specific. Also the forensic investigation depends on the special circumstances in the robots environment. So, the focus of the prototype is recording the received state data as shown in section III to generate a log-file that contributes to a forensic investigation. We chose embedded (no operating system) hardware with high computing power that supports the use of cryptographic functions and runs pre-compiled source code "on the metal" to minimize the attack surfaces. Therefore, all the operating system related vulnerabilities and exploits can be eliminated.

The Atmel® SMART SAMA5D2 series with high-performance, ultra-low-power ARM® Cortex®-A5 processor-based MPU (Micro Processing Unit) running up to 500MHz fits our purpose. The device integrates powerful peripherals for connectivity and offers advanced security functions (ARM TrustZone®, tamper detection, secure data storage, etc.) as well as high-performance crypto-processors for AES, SHA and TRNG (True Random Number Generator).

The communication between the controller and the BB is done via an USB 2.0 link, which enables a symbol rate of 480 Mbps. All the data, which must be stored permanently over a long time period, is written to a connected SD card.

The performance of asymmetric cryptographic operations is especially important in our application. Using a platform-specific crypto library, the performance values shown in table I can be achieved. The use of the Chinese Remainder Theorem (CRT) [19] allows a speedup of RSA operations through faster calculation of the underlying mathematical functions while slightly diminishing the level of security. We decided against the use ECDSA [20] for signature creation and validation in sake of the RSA algorithm since this provides a trade-off between higher security and higher performance.

According to table I, we can sign 12.72 hash values of 1024 bits per second (a hash value is the representative for a log file entry) without applying the Chinese Remainder Theorem or 37 entries per second with CRT respectively. Here, it can be seen that the asymmetric cryptography is a bottleneck. This is why we have foreseen the possibility to not sign every single block entry but rather skip some entries and sign every n^{th} entry depending on the incoming data frequency.

Operation	CPU Cycles	Timing per block
RSA1024 $S(m, sk)$ w/o CRT	39.3 MCycles	78.6 ms
RSA1024 $S(m, sk)$ w. CRT	13.5 MCycles	27 ms
RSA2048 $S(m, sk)$ w/o CRT	270 MCycles	540 ms
RSA2048 $S(m, sk)$ w. CRT	79.2 MCycles	158.4 ms
RSA4096 $S(m, sk)$ w/o CRT	2004 MCycles	4008 ms
RSA4096 $S(m, sk)$ w. CRT	540 MCycles	1080 ms

TABLE I
PERFORMANCE OF RSA ALGORITHM WITH THREE DIFFERENT BIT BLOCK SIZE (1024, 2048 AND 4096).

The connection to the controller is not a limiting factor here. Obviously, it is not feasible to store all data, which a robotic system produces during its operation (a very extreme example is the amount of data collected by Google's self driving car, which reaches 1GB/s[2]). However, our prototype provides enough storage bandwidth to store meaningfully abstracted data. Instead of raw sensor data, sensor results (e.g., detected objects) and associated confidences already provide significant evidence. In addition, as argued in [1], also the decisions of a planning component can easily be included in the available processing bandwidth.

As an example, let us consider a mobile robot with a differential drive and two LIDAR sensors (e.g., Sick S100). Assuming we want to record the state of the system at 25 Hz and we record the current speed of each driven wheel (2 times 2 bytes) and the detections of the LIDAR (assuming an angular resolution of 1° and an aperture angle of 270° with 2 bytes per degree). At 25 Hz, this will produce 13600 bytes per second but without CRT (12.72 signatures per second) only every second block entry can be signed. This means that if a digital signature can not be verified, the authenticity of two block entries can not be guaranteed and 1.06 kB of data is lost. When applying the CRT (37 signatures per second) it is possible to sign every block entry. Remember, signing a block entry means signing its hash value and not the original data itself. The stored data provides enough meaningful data to reconstruct the situation of the robot with a temporal resolution of 40 ms.

[2]http://www.kurzweilai.net/googles-self-driving-car-gathers-nearly-1-gbsec

VI. Conclusion

Using the requirements defined in section III-A, we discuss the possibilities and limits of our approach.

Requirement *R1* is achieved by recording state data in a physically separated device, which should also be specially protected from physical damage. Reliable storage of data is ensured by digital signatures, which enable the detection of data loss or tampering. The use of a hash-chain ensures that data is stored in the correct order.

Referencing *R2*, confidentiality, integrity and authenticity are achieved by encryption, hashing and signing of data. Strong encryption ensures that data cannot be read by unauthorized parties. The hash chain ensures that consecutive data rows are stored in the right order, that no data is inserted, deleted or tampered with. The digital signature can be used to verify that the stored data has been written by an authentic data recorder and that no data has been maliciously inserted.

To increase tamper resistance (requirement *R3*), the hardware board features special tamper pins, which can be connected to the housing. In case it is opened without activating the maintenance mode, this can be detected and countermeasures are activated. In addition, as soon as the constant data stream from the controller stops, drops significantly in frequency or exceeds defined threshold values, the bionic approach will kick in and execute the pre-defined rules which are meant to ensure safety and protect the system from damage and tampering.

To use the collected data in forensic investigation, it is necessary, that each significant event is recorded on the BB. This must be ensured by the system diagnosis of the controller. However, the BB ensures, that each entry of the log is kept safe and that its integrity is preserved. Since our concept defines clear workflows for the collection, processing and storage of data, this can eventually be certified and will thus suffice in producing forensic evidence.

VII. Further Work

To further extend our concept, we plan several improvements in our future work. We want to develop smart reconstruction mechanisms, which are able to partially verify signatures despite loss of data. Since a signature cannot be verified if only a single bit of data flips, already limited damage to memory are problematic. If however, a signature scheme can provide some confidence on how likely data has been tampered with, this could be beneficial to forensic causes.

We plan to add more scalability to our black box implementation ranging from software-only imlementations to high-performance hardware-boxes.

Finally, we will extend our conceptual basis towards more transparency for inexperienced users. The goal is that users can gain insight into the inner workings of a robot. Here, the black box can act as an interface, which provides information on what is currently happening in the system, which kinds of data are collected and stored and what the specific application of a robot is.

Acknowledgements

The work reported in this article has been supported by the Austrian Ministry for Transport, Innovation and Technology (bmvit) within the project framework Collaborative Robotics and by the programme "ICT of the Future", managed by the Austrian Research Promotion Agency (FFG), under grant no. 861264.

References

[1] A. F. T. Winfield and M. Jirotka, "The case for an ethical black box," in *Towards Autonomous Robotic Systems*, Y. Gao, S. Fallah, Y. Jin, and C. Lekakou, Eds. Cham: Springer International Publishing, 2017, pp. 262–273.

[2] R. H. Wortham, A. Theodorou, and J. J. Bryson, "What does the robot think? transparency as a fundamental design requirement for intelligent systems," in *Proceedings of the IJCAI Workshop on Ethics for Artificial Intelligence*, June 2016. [Online]. Available: http://opus.bath.ac.uk/50294/

[3] A. Theodorou, R. H. Wortham, and J. J. Bryson, "Why is my robot behaving like that? designing transparency for real time inspection of autonomous robots," in *AISB Workshop on Principles of Robotics*, April 2016. [Online]. Available: http://opus.bath.ac.uk/49713/

[4] "European parliament resolution of 16 february 2017 with recommendations to the commission on civil law rules on robotics (2015/2103(inl))," Reference: P8-TA-2017-0051, Feb. 2017.

[5] D. R. Grossi, "Aviation recorder overview," in *International Symposium On Transportation Recorders, Arlington, Virginia*, 1999.

[6] D. R. Warren, *A device for assisting investigation into aircraft accidents*. Aeronautical Research Laboratories, 1954.

[7] J. B. Groenewegen, "Crash survivable enclosure for flight recorder," Jul. 31 1990, uS Patent 4,944,401.

[8] ——, "Heat shielded memory unit for an aircraft flight data recorder," Sep. 15 1987, uS Patent 4,694,119.

[9] G. Walker and A. Strathie, "Leading indicators of operational risk on the railway: A novel use for underutilised data recordings," *Safety Science*, vol. 74, pp. 93 – 101, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925753514003051

[10] S. Jambhekar, J. Hara, and J. Barr, "Method and device for vehicle control events data recording and securing," Jun. 13 2000, uS Patent 6,076,026. [Online]. Available: https://www.google.com/patents/US6076026

[11] K. Noguchi, "Data recording apparatus and the method thereof," Sep. 1 2005, uS Patent App. 11/065,069. [Online]. Available: https://www.google.com/patents/US20050190468

[12] S. Choudhuri and T. Givargis, "Flashbox: A system for logging non-deterministic events in deployed embedded systems," in *Proceedings of the 2009 ACM Symposium on Applied Computing*, ser. SAC '09. New York, NY, USA: ACM, 2009, pp. 1676–1682. [Online]. Available: http://doi.acm.org/10.1145/1529282.1529657

[13] S. Elbaum and J. C. Munson, "Software black box: an alternative mechanism for failure analysis," in *Proceedings 11th International Symposium on Software Reliability Engineering. ISSRE 2000*, 2000, pp. 365–376.

[14] NIST, "Advanced encryption standard (aes)," 2001, 'FIPS PUB 197'.

[15] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120 – 126, 1978.

[16] NIST, "Secure hash standard (sha)," 2002, 'FIPS PUB 180-2'.

[17] W. W. Peterson and E. J. Weldon, *Error-correcting Codes*. The Massachusetts Institute of Technology, 1972.

[18] K. D. Roeder, L. Tozian, and E. A. Weiant, "Endogenous nerve activity and behaviour in the mantis and cockroach," *Journal of Insect Physiology*, vol. 4, no. 1, pp. 45 – 62, 1960. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0022191060900676

[19] G. A. Jones and J. M. Jones, *Elementary Number Theory*. Springer-Verlag, 1998.

[20] IETF, "Deterministic usage of the digital signature algorithm (dsa) and elliptic curve digital signature algorithm (ecdsa)," 2013, 'RFC 6979'.