Proposal for a Secure Forensic Data Storage

Nico Vinzenz[1,*] and Tobias Eggendorfer[2]

[1]*ZF Friedrichshafen AG, 88046 Friedrichshafen, Germany*
[2]*Hochschule Ravensburg-Weingarten, 88250 Weingarten, Germany*
Email: nico.vinzenz@zf.com; tobias.eggendorfer@hs-weingarten.de
[*]*Corresponding Author*

Received 14 April 2020; Accepted 12 July 2020;
Publication 30 October 2020

Abstract

A forensically sound and secure data storage architecture is proposed in this paper. A design focus was on its tamper-proof and memory saving, yet cost-efficient ability to store information valuable for both post-crash and post-crime investigations in a privacy protecting manner. Current privacy regulations were extensively taken into consideration by its architecture design. The implementation requires only minor changes to the vehicle software architecture, leaving the vehicular hardware completely untouched.

Keywords: Secure forensic data storage, vehicle forensic, vehicle security, legal car application, crash investigation, event data recorder, privacy, privacy preserving forensics.

1 Introduction

The investigation of vehicular data stores is a powerful method for reconstructing both crash and crime related incidents. In the past, Airbag Event Data Recorders (EDR), Diagnostic Trouble Codes (DTC) or even the telematics unit of a car were analyzed for its data content [1]. However, the

Journal of Cyber Security and Mobility, Vol. 9_3, 469–488.
doi: 10.13052/jcsm2245-1439.934

amount of data which is being processed by modern vehicles has risen drastically in recent years. This development is fueled by steady increasing interconnectivity requirements and the integration of much more powerful controllers.

Most of this data is not kept persistently within the vehicle. A privacy preserving, however forensically sound and memory efficient method to store this variety of data streams in a secure, tamper-proof manner would strongly assist a subsequent forensic investigation.

In this paper, a secure forensic data storage which addresses these challenges is proposed.

2 Related Work

Various publications such as [2] and [3] have already proposed optimizations in vehicular architecture to construct a data recorder for a forensic analysis. However, they require changes to the hardware architecture within the car to achieve their objectives, which is hard to realize, since car manufacturers rely heavily on Original Equipment Manufacturers (OEM) as well as having to adhere to legal and standard requirements, and due to mass production, need to keep unit costs as low as possible. Unlike these publications, this paper's proposal only requires the software architecture to be adjusted and leaves the hardware untouched, thereby reducing implementation costs and maintaining compliance to existing architectures.

3 Proposal for a secure forensic data storage

The main objective of this proposal is to facilitate a forensic investigation by providing the best possible obtainable dataset of a vehicle. Security, technical feasibility, privacy concerns as well as possible future online extensions are thereby addressed with special consideration.

Initially, Section 3.1 stipulates the requirements this architecture proposal must fulfill. Section 3.2 describes the architecture proposal and Section 3.3 evaluates it against the requirements.

3.1 Requirements on the architecture

To achieve a high level of maturity there are several requirements the forensic architecture proposal must fulfill:

1. **Forensic information value:** The proposal must allow getting an insight into the past vehicle state to a degree it would not be possible with regular forensic methods. It makes only sense to change the vehicle architecture if there is actual value added for a forensic investigation.
2. **Authorization:** All stored data can only be accessed by an authorized investigator through an authentication procedure. The recorded personal data is subject to the General Data Protection Regulation (GDPR) and must therefore be protected from unauthorized access and improper use.
3. **Confidentiality:** All stored data must be protected from disclosure to unauthorized parties. This protection must even hold true from a party which is authorized to access the vehicle controllers, e.g. a service technician. Like the authenticity requirement, confidentiality is necessary to fulfill GDPR requirements for personal data.
4. **Authenticity:** All stored data is verifiable not tampered nor corrupted (integrity). The data must stand up in a court case and furthermore fulfill the right of correctness (GDPR Art. 5 (I) lit. d)).
5. **Privacy:** In addition to the data protection requirements specified with authenticity, confidentiality and integrity, any other privacy requirements stipulated by the GDPR legislation must be fulfilled.
6. **Non-repudiation:** No one can successfully dispute the validity of the stored data. This is necessary such that the data can stand up in court.
7. **Availability:** All stored data can be accessed in a timely manner. A time critical investigation requires fast access to the data and cannot be impaired by a tedious extraction process.
8. **Full offline functionality:** The proposal is not dependent on an Internet connection to a server but works completely offline. While full mobile coverage is not guaranteed, the forensic data storage cannot be dependent on an Internet connection. Furthermore, online functionality would require additional server infrastructure as well as increasing the complexity to maintain privacy.
9. **Technical feasibility and scalability:** The proposal must take technical limitations into account, such as CPU calculation cost, flash wear and data size limitations. Furthermore, it should be scalable in hardware requirements. This allows the deployment on not only a specific, but a range of devices with different hardware configurations.
10. **Cost effectiveness:** The proposal cannot contain expensive elements such as hardware modifications. In the automotive industry vehicles are produced in high quantities and even a slight unit cost increase leads to a high rise of total expenses.

3.2 Architecture proposal

Before defining how the secure forensic data storage architecture operates within a vehicular environment, it is important to decide on an adequate location. Ideally, the device where it is deployed already connects various information streams such as vehicle bus communication (CAN, Ethernet), GPS, mobile broadband, WiFi, Bluetooth, etc. by design. Having this prerequisite, data can be recorded as needed and does not require any traffic flow forwarding, which would increase complexity, bus load and latency. Because the data recording puts additional requirements on CPU performance, as well as memory speed and size, a desirable device requires also significantly better hardware than regular ECUs. With these constraints, the only reasonable choice for deploying a secure forensic data storage is the central communication hub within the vehicle – the telematic platform.

However, simply implementing a write-through of all data streams into the telematic platforms flash storage is not enough to fulfill the requirements of Section 3.1. The available flash storage would rather quickly be filled up with all sorts of data, making it unable to store new forensic evidence. Not only is space wasted by storing partially non-relevant information, but there is also a method lacking to ensure data integrity.

A more sophisticated approach is to define a dataset out of the numerous available data streams, containing only forensically relevant data. They are then bundled together into a data block. The available data streams are different for each vehicle hardware setup. In the best case, a data block will contain information valuable for both post-crime and post-crash investigations, such as GPS data. It is then compressed and written into a flash storage circular buffer, such that the oldest entry is overwritten when it is full. Unfortunately, this approach does not enforce an efficient data protection whatsoever.

A possible solution to protect the data's confidentiality would be encrypting it with a symmetric cipher. However, if the device is compromised, e.g. by a privilege escalation, then the attacker can decrypt all information. Even without a direct attack this approach is insufficient. For instance, a service technician may require the root credentials to work on the device but should never be able to decrypt and read the recorded data. Potential leakage of the symmetric key during device production is another risk.

These problems are solved with asymmetric cryptography. For this approach, the manufacturer creates vehicle specific public and private key pairs. The private key is exclusively stored in a company KMS (Key Management System) which acts as a key escrow. On the contrary, the associated

public key is inserted into the telematic platform during vehicle production. Within the vehicular scope, applying a public key operation resembles a one-way transformation which is infeasible to reverse without knowledge of the private key. The private key is only handed over by the company to an investigator in case of a justifiable reason, which must be backed up with a court order. However, asymmetric algorithms such as RSA or ECC are generally not suitable for encrypting large data volumes. RSA can only encrypt small data chunks with low performance and ECC cannot be used to encrypt data altogether.

This problem is solved by combining both, symmetric and asymmetric cryptography, into a hybrid encryption scheme. The most trivial construction would use a CSPRNG (Cryptographically Secure Pseudorandom Number Generator) generated key for encrypting large data volumes with a symmetric cipher such as AES-CBC (Cipher Block Chaining). The symmetric key would then be encrypted with RSA and the public key.

An even better construction, and therefore used in this architecture proposal, is provided by the standardized ECIES (Elliptic Curve Integrated Encryption Scheme). The ECIES is based on two parts: (1) ECDHE (Elliptic Curve Diffie-Hellman Ephemeral) key agreement protocol and (2) AE (Authenticated Encryption). For implementing the ECDHE, in addition to the static ECC public/private key pair (split between telematic platform and KMS), a second ephemeral key pair is generated locally on the telematic platform for each authenticated encryption. The main idea is now to use the ephemeral private key for generating a shared secret with the static public key, which is then fed into a KDF (Key Derivation Function) to finally generate the symmetric key. The ephemeral private key is purposely discarded afterwards, whereas the ephemeral public key is appended to the ciphertext. The same secret can only be calculated again when the attached key is combined with the static private key from the key management system. Using ephemeral keys gives the advantage of forward secrecy, meaning that even when an attacker can get the symmetric encryption key for the current encryption operation, ciphertext of previous data blocks cannot be decrypted. The upside of this construction compared to RSA is a much higher performance and smaller key size.[1]

[1]Own measurement: on an ARM Cortex-A53 1.2 GHz CPU without hardware cryptography support, ECDH is 6 times faster compared to RSA key encryption. A 256 bit security level can achieved by either a 512 bit ECC key or a 15360 bit RSA key, which is 30 times larger.

The generated symmetric key is then used for encrypting the data with an AE algorithm such as AES-GCM (Galois/Counter Mode). This algorithm produces, additionally to the ciphertext, an authentication tag which protects both integrity and authenticity of the data block. A block-chain is created by appending the predecessor tag to a data block plaintext before the AE algorithm is applied to it, effectively making the newly generated tag dependent of all preceding tags (see Figure 3). The concatenation of ciphertext, tag and public key is then finally inserted into the circular buffer.

The detailed process for storing forensic data is described in Section 3.2.1 and its retrieval in Section 3.2.3.

3.2.1 Storing forensic data

Figure 1 shows the proposal for proactively storing forensic data inside a vehicular telematic platform.

In the first step, a data block DB_n is created in volatile memory and filled with information from available data streams. Depending on the vehicle hardware configuration, this information may include position data (GPS, cell site ID), passenger information (via Bluetooth/Wi-Fi connected devices), vehicle status data (CAN/Ethernet messages, DTCs), etc. Every vehicle model is different; therefore a vehicle model specific implementation is required.

Atomic data elements, i.e. GPS point, DTC, Bluetooth status information, etc. have a fixed format and size within the DB_n. They are encoded by a TLV (Type-length-value) scheme such as ASN.1 DER. A sample rate defines how often the most recent value for each data element is put into the DB_n. For example, the GPS module could provide 10 GPS points per second, where every 500 ms the most recent value is sampled.

Each DB_n is filled for a predefined timeslot, e.g. 1 s. When this timeslot passed, DB_n is closed and the data stream redirected to its successor block. The amount of sampled data is not completely deterministic because irregularly generated data such as DTCs are also stored, leading to a variable DB_n size. To prevent a DB_n from getting indefinitely big, an upper size bound is defined and data elements with the lowest priority are skipped for this timeslot.

In the next step, DB_n is compressed, for instance with the LZMA algorithm, yielding $Compr(DB)_n$ and forwarded to the HSM (Hardware Security Module) or TEE (Trusted Execution Environment) section of the device (if available).

Within this environment, the X25519 key agreement which is based on ECDHE using Curve25519 is performed. Prerequisite is the static B_{pub}

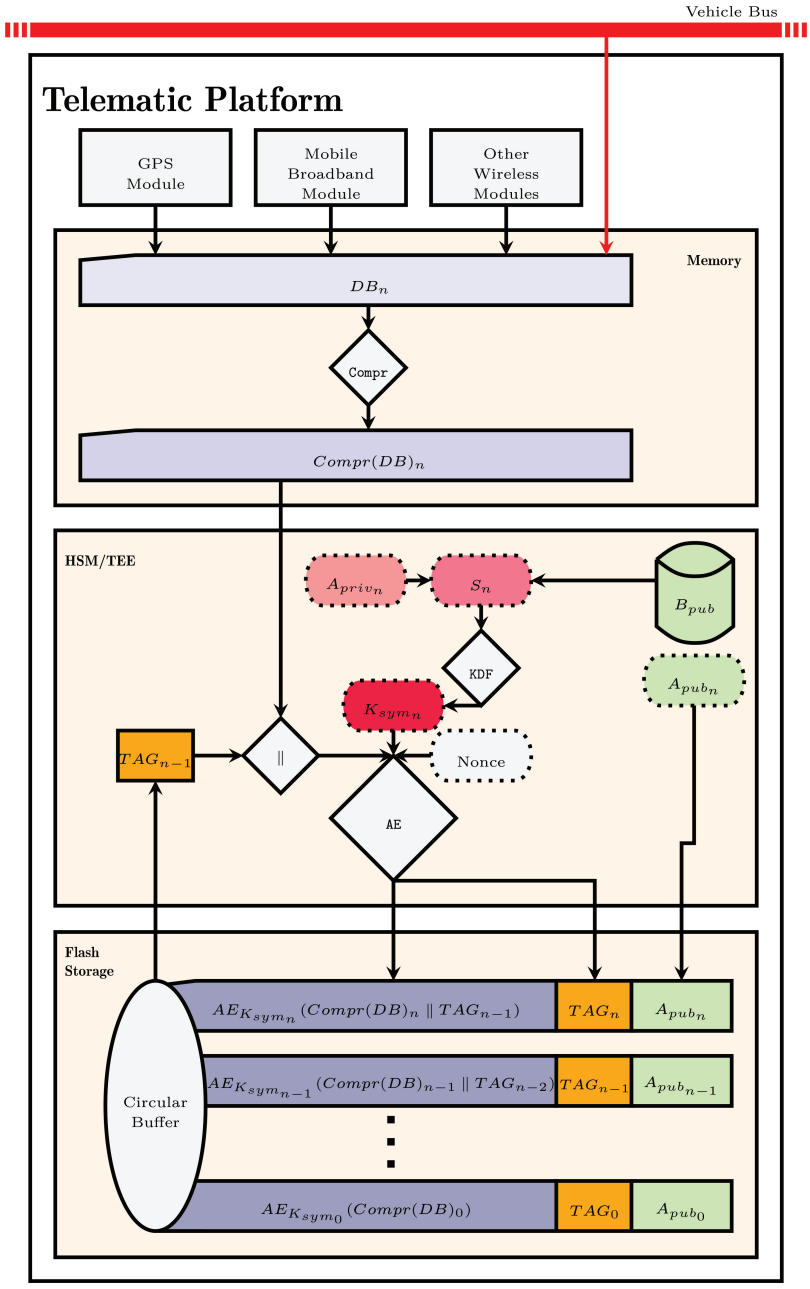


Figure 1 Schematic illustration of the proposed architecture for securely storing forensic data.

which is vehicle-specific and inserted during device production, in best case permanently burned into fuses. First, the ephemeral key pair (A_{priv_n}, A_{pub_n}) is generated for one-time use. A curve multiplication of A_{priv_n} with B_{pub} results in the shared secret S_n. The hash-based $KDF(S_n)$ is then calculated to remove weak bits from S_n, producing K_{sym_n}.

If there is already data within the circular buffer, the TAG of the most recent element is read and concatenated, yielding $Compr(DB)_n \parallel TAG_{n-1}$. This result is then used together with K_{sym_n} and the current index $n = Nonce$ as input for the AE algorithm. The AE output is $AE_{K_{sym_n}}(Compr(DB)_n \parallel TAG_{n-1})$ as well as TAG_n.

This output is bundled with the ephemeral A_{pub_n} and put into a circular buffer CB within the flash storage on index n, that is $CB_n = AE_{K_{sym_n}}(Compr(DB)_n \parallel TAG_{n-1}) + TAG_n + A_{pub_n}$. Even though CB is limited to a fixed size, it will never block a writing operation because the oldest entry is overwritten when it is full. See Section 3.2.2 for a detailed implementation description of the circular buffer.

At the end of each encryption, A_{priv_n} is deliberately overwritten, making a reconstruction of $K_{sym_n} = KDF(A_{priv_n} * B_{pub})$ impossible.

3.2.2 Implementation details of the circular buffer

The circular buffer CB residing within the flash must have a fixed maximum size, otherwise it would eventually completely fill up the available flash storage, leading to a non-operational telematic device. The content without overhead of each entry within CB contains for all indices $n > 0$ the format $CB_n = AE_{K_{sym_n}}(Compr(DB)_n \parallel TAG_{n-1}) + TAG_n + A_{pub_n}$. An exceptional case is the first entry with index $n = 0$ having the format $CB_n = AE_{K_{sym_n}}(Compr(DB)_n) + TAG_n + A_{pub_n}$.

The size of each entry is variable with an upper bound. This bound guarantees a information coverage for a defined minimum time period until CB must overwrite old entries. The worst case information coverage time span is determined by the time slot of each entry, additionally assuming the max bound size for each entry and a data compression ratio of 1:1. For example, having a 512 MB circular buffer with an upper entry bound of 128 KB (including overhead of 64 B) and a time slot of 1 s would provide at worst 1 h 6 m 40 s of information. If vehicle power is instantly cut off during a crash in worst case the most recent 1 s of data could be lost.

The circular buffer can be implemented in multiple ways. Because of the variable entry size, it is not appropriate to use a static array-like data structure. A better method is using a singly linked list implementing a modified

LIFO (Last-In First-Out) circular buffer variant. Another upside is its high performance of $\mathcal{O}(1)$ for element insertions and reads.

For the proposed implementation, first an empty file of maximal available size is generated within the telematic platform. It is essential that this file can only be edited by a designated user specifically created for this task, in best case only by the HSM or TEE.

Some overhead is required for storing CB entries. A pointer $last\_stored\_ptr$ is put at the file beginning, pointing to the last stored entry (for the empty file that is $file\_start + last\_stored\_ptr\_size$). All entries contain the fields $index$, $predecessor\_pointer$, $length$ and $value$. The field $index$ starts with 0 and is incremented for each entry, giving an overview how many entries were already created in total and also serving as $nonce$. The $predecessor\_pointer$ points to the previous entry start address. Finally, $length$ denotes the length of $value = AE_{K_{sym_n}}(Compr(DB)_n \parallel TAG_{n-1}) + TAG_n + A_{pub_n}$.

Based on this information the position of the last stored entry as well as all predecessors is known. A new entry is now put adjacently to the last stored entry. A special case is when the to-be-stored entry requires more space than is remaining within the file. In this case, first the remaining space is exhausted until $file\_end$, and then writing is continued at position $file\_start + last\_stored\_ptr\_size$, overwriting the data residing there. From that point on, the oldest entries are overwritten by newer entries.

To read all entries from CB, first the most recent element's position is determined by $last\_stored\_ptr$. All predecessors can then be traversed until the end is reached and $predecessor\_pointer$ points to an invalid location.

3.2.3 Retrieving forensic data

Figure 2 shows the proposed process for a forensic investigator to retrieve the preserved data from a vehicular telematic platform.

Initially, the circular buffer CB file is extracted from the flash storage of the telematic device to a forensic workstation. If the device is still in working condition, then an investigator can connect with company provided credentials for simple extraction. In case this is not possible, due to damage or other reasons, the file may be extracted from the flash with chip-off forensics.

All elements $CB_{n..0} = AE_{K_{sym_n}}(Compr(DB)_n \| TAG_{n-1}) + TAG_n + A_{pub_n}$ are readout by traversing CB from the most recent (denoted by n) to the oldest entry.

First, for all elements $CB_{n-1..0}$ the appended tags $TAG_{n-1..0}$ are verified to be bit-identical with those tags $TAG_{n-1..0}$ extracted from their successor's

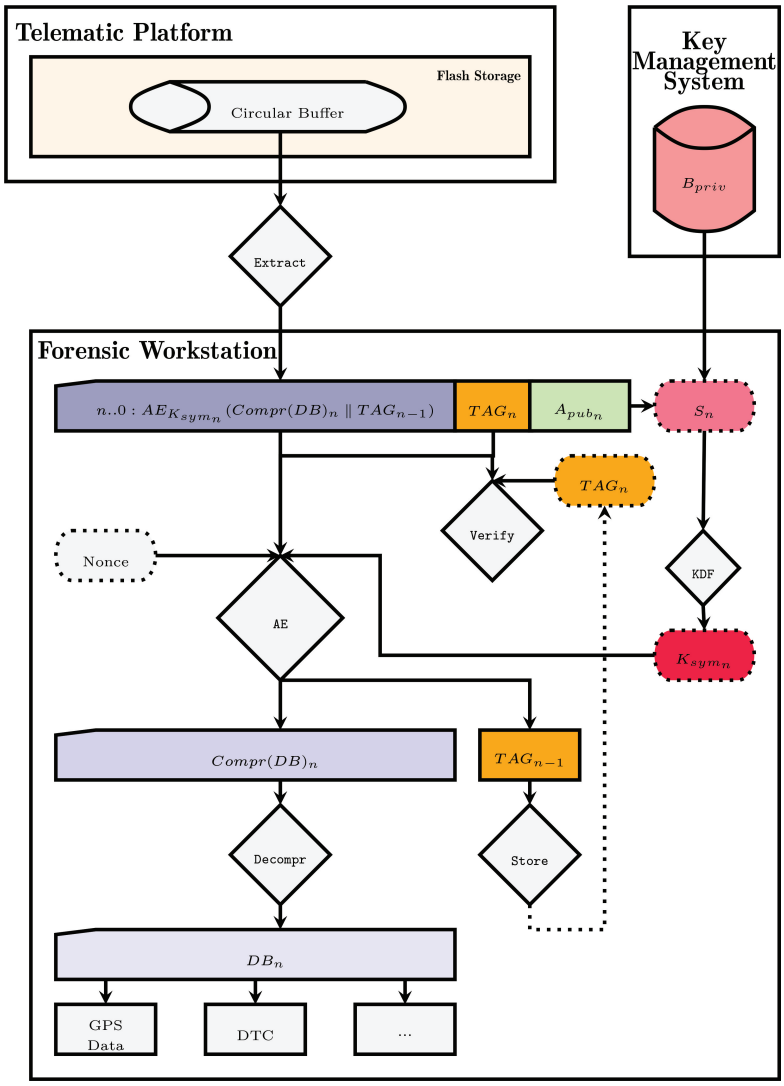


Figure 2 Schematic illustration of the proposed architecture for retrieving and verifying forensic data.

CB_n payload (see Figure 3). If there was any adversarial insertion within the chain, it will be detected by this verification step. Because the blockchain is traversed backwards, this verification step is possible even when entries were overwritten within the circular buffer.

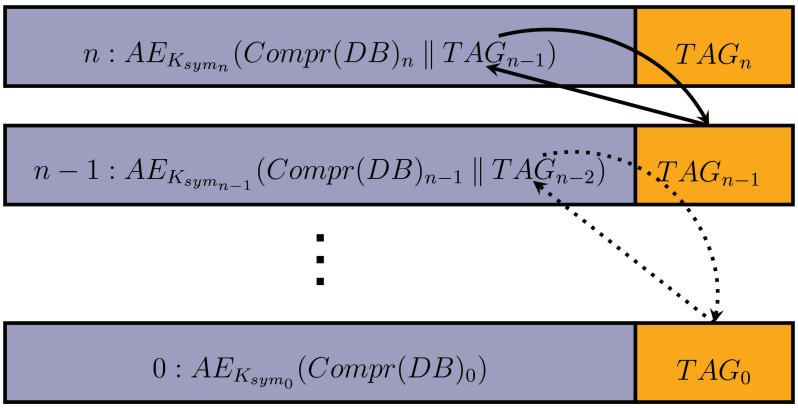


Figure 3 A blockchain mechanism is achieved by including a message TAG into each successors ciphertext. Traversing the chain backwards allows for checking the chain consistency.

Next, for each element $K_{sym_n} = KDF(A_{pub_n} * B_{priv})$ is reconstructed. This is only possible when the investigator is granted access to B_{priv} stored within the companies KMS. A_{pub_n} is obtained from the element itself.

The AE algorithm is now executed with the input tuple $AE_{K_{sym_n}}(Compr(DB)_n \parallel TAG_{n-1}), TAG_n, Nonce = n$ and K_{sym_n} for decryption. If there was any bit-change in the ciphertext, it will be recognized in this step based on the authentication tag.

The decrypted TAG_{n-1} is stored for the verification process of the next element, as explained above. Finally, $Compr(DB)_n$ is decompressed into DB_n, which is then split into its atomar data elements GPS data, DTCs, etc.

3.3 Requirement fulfillment

This section discusses how each of the requirements specified in Section 3.1 is fulfilled.

Forensic information value is added with this architecture proposal by utilizing the telematic platform to collect information from multiple data streams such as CAN, Ethernet, GPS, wireless networks, etc. If each second 200 byte of compressed data is collected, then a whole month of vehicle operation time can be preserved in approximately 520 MB flash storage. For example, at such a sample rate around 7 GPS data points in NMEA 0183 format [4] can be stored. This diversity and long time period of continuously collected information is beneficial for both post-crash and post-crime investigations.

Authorization and confidentiality are achieved with the hybrid usage of symmetric and asymmetric encryption. Even if the device is fully compromised, it is not possible to decrypt any data. For accessing the forensic data an authorized access to the private key from the company key management system is necessary. This key should only be handed over to an investigator on basis of a justifiable reason backed up with a court order.

Authenticity, integrity and non-repudiation are accomplished with the authenticated encryption scheme. Both tampered and corrupted data is recognized during decryption, making it also infeasible to dispute their origin. Block-chaining the message tags hardens the implementation even further, allowing the detection of correctly calculated adversarial data blocks within the chain.

See Section 5 for an elaboration on data protection and privacy within an vehicular context.

Availability is achieved by the proactive forensic measure to write forensically useful data streams persistently in flash storage and make it accessible to an investigator. The retrieval process can be automated and takes, even for multiple GB of forensic data, only a couple of minutes with a fast workstation.

Full offline functionality is guaranteed by design. None of the implemented functions require an online connection.

The architecture proposal is technically feasible and scalable because it utilizes available vehicle components. It can be adapted to different hardware setups individually. If the storage process is too calculation intensive for the device, then the time slot for a data block can be increased and the sample rate of the data streams decreased. The same applies to the size of the storage space which can be scaled to available resources. However, the storage of each data block is lightweight and only consists of a compression and authenticated encryption operation. The ECDHE key agreement is fast and can be applied for each single auth-encrypt again without causing a relevant increase in latency.[2] Telematic devices which are designed to process large data volumes every second can handle this extra workload. The hardware cryptography support provided by most newer architectures accelerates the storage process even more.

[2]Own measurement: on an ARM Cortex-A53 1.2 GHz CPU with 1 GB memory and without hardware cryptography support, LZMA compression and AES-GCM take on average approximately 201 ms to complete for an 100 kB data block. ECDHE key agreement takes about 2 ms.

The proposal is cost effective because it utilizes the information available within the telematic platform scope. It does not require any kind of hardware modification. Newer telematic platforms have a secure key storage and hardware cryptography support already included by default. A key insertion at production time, as well as a company key management system, are both necessary either way to manage credentials for general device access. Furthermore, the needed software modifications are mainly based on cryptographic primitives which can be implemented with available libraries.

4 Security Limitations

This proposal has a few security limitations in case the telematic platform is completely compromised by an attacker. A complete compromise means that the attacker obtained root privileges and can extract the keys from the HSM or TEE key vault.

With root privileges the attacker could also delete every data block in the flash storage. More severe, he could manipulate the data streams such that new data blocks with forged forensic data are created. By knowing the public key B_{Pub}, own forged data blocks could be created. These attacks can be mitigated with an online extension and is discussed in Section 7.

Another attack vector is the company key management system. Unauthorized access to both root credentials and the private key B_{Priv} will allow the above described attacks and the complete disclosure of all forensic data within the flash. A mitigation for credential leaks or the malicious use by company-intern credential holders is accomplished by using non-root accounts as well as applying HSM/TEE isolation techniques on the device. The user privileges of each account must be tailored to the respective role. Root credentials should preferably not be stored.

5 Data Protection

Data collected in vehicles allows for very precise analysis of a person habits: From travelling times and destinations, via driving style and aggressivity, up to how many people usually travel in the vehicle thanks to seat belt sensors or music choices. All of this data is highly personal - when German car insurances suggested to use this data to offer a rebate to safe drivers, a public outcry was the result.

To protect personal data the European Union has provided a unified regulation: General Data Protection Regulation, GDPR. Several nations outside

the EU have successfully applied to have their data protection laws considered to be adequate by the EU commission [5, 6], while e.g. California is updating their state data protection laws to EU standards. Therefore, it seems legitimate to use EU-GDPR as a base to analyse requirements for forensic data storage and the usage of this data.

EU-GDPR does not provide for forensic data logging by itself, however, the regulation allows data processing in Art. 6 (1) to fulfill legal requirements as well as for legitimate purposes, as long as they are weighed against the impact on the individual, to fulfil contracts, protect from life-threating effects on the individual or others, or, simply by the individual's agreement.

Albeit released prior to GDPR being ratified, [7, p. 31] notes that a strict privacy policy "contrasts with other rights or obligations, including the interest in effective road safety research [..], ensuring that roads are used without infringing the highway code, third party rights (e.g. victims' rights), driver interests, and adherence to a contract with another entity."

5.1 Data processing entity

5.1.1 Car owner

GDPR also needs the data processing entity to be identified. In car forensics there might be different setups: If data is kept in the car only and only accessible by the car's owner, the car's owner is the data processor. If the car is owned by a natural person, GDPR is not applicable (GDPR Art. 2 (II) lit. c). If the car is owned by a rental company or any other business, the respective business owner has to take adequate precautions to protect this data.

5.1.2 Manufacturer

If, however the manufacturer is - as most current schemes suggest – the only one able to access the data, the manufacturer becomes the data processor, no matter whether data is stored in car, in a cloud or on a server at the manufacturer's. This results in all GDPR obligations needing to be met, from providing adequate data security measures to being able to fulfill a data subject's rights such as access to data, correction, deletion, restriction of processing as well as data portability, with all of these entangling complex legal questions.

Also the manufacturer would have to identify a plausible cause for this data processing: Lacking a legal requirement this could either be by contract,

providing preventive forensics as a service to be prepared in case of an accident, or by the customers consent, which however could be withdrawn at any time.

5.1.3 Governmental body

A government could by law enforce manufacturers to store forensic data in a government run cloud, where only governmental institutions would have access. Then, data collection was a legal requirement, however the manufacturer would not have access to this data. Since any concept requiring new legislation would be beyond the scope of this paper, it is left for future work.

5.1.4 Multiple entities

Besides the manufacturer, also the dealer ship, a garage or other service agents could have access to the forensic data. If they do, from a legal perspective clarification is needed as to whether they act on their own or on the manufacturer's behalf. The latter reducing the issue to the manufacturer being the sole data processor, whereas the former results in a joint controller situation.

Also, a joint controller situation could arise between a commercial owner, such as a rental car agency, and the manufacturer, if both had access indepently to the forensic data.

5.1.5 Suggestion

In order to keep complexity low, provide an optimal level of data protection by not sending it over networks, it seems reasonable to store data in the vehicle. This is in accordance to a 2014 study [8] (again) predating the GDPR, prepared for the European Commission regarding the installation of EDRs in vehicles, where the German Federal Data Protection Authority commented on the minimum requirements for vehicle information collection to be lawful.

5.2 When and what data to store

The same study also provides reasonable guidance on when to store data:

They state that only vehicles for the transport of dangerous goods and buses should have mandatory data recorders. Without explicit consent of the data owner it is only acceptable to have event triggered and not continuous data logging implemented. The data owner should further be able to turn

the data logging on and off at any time, if not otherwise specified by a contract. Each stored data element must be transparent for the data owner and should not be stored centralized on a server but decentralized within the vehicle. Finally, all personal data must be protected against fraudulent use by encryption and access control. The set of data which each party (crash investigator, service technician) can access must be clearly defined and any additional access is prohibited.

5.3 Architecture adaption to data protection regulations

Depending on how courts and legislature will decide on the trade-off between the public interest in transport network safety and the protection of personal data in the future, three variants of implementing the architecture proposal exist.

5.3.1 Variant 1

The first variant is to leave the architecture proposal of Section 3 unchanged. However, it is important to meet GDPR principles of processing personal data defined in GDPR Art. 5. The data owner must be informed which data is stored (GDPR Art. 5 (I) lit. a)). Only data in relation with a post-crash and post-crime investigation can be recorded (GDPR Art. 5 (I) lit. b)). Moreover, this data can only be processed for this purpose (GDPR Art. 5 (I) lit. c)). The correctness (GDPR Art. 5 (I) lit. d)), integrity and confidentiality (GDPR Art. 5 (I) lit. f)) of the data is safeguarded by the architecture by default. Furthermore, the circular buffer with automatic deletion of old data ensures that data is stored no longer than necessary (GDPR Art. 5 (I) lit. e)).

5.3.2 Variant 2

The second variant requires a change of the architecture proposal. The "nemo tenetur se ipsum accusare"-principle or "right to silence" states that nobody is bound to self-incrimination [8, p. 176]. If courts or legislature decide that the recording of forensic data is a violation of this principle, then an additional protection layer must be added to the proposal. This layer prevents prosecution from accessing the forensic data of its owner. It could be implemented with an additional password which is only known to the data owner. This password is concatenated with the temporary symmetric key and hashed, creating the new symmetric key which is then used to encrypt the data blocks. With this method each data owner can decide for themselves if they want to disclose their data.

5.3.3 Variant 3

The last variant is to refrain from implementing an architecture for forensic data collection. This is necessary if courts or legislature conclude that the storage of forensic data is incompatible with data protection regulations.

5.3.4 Results

Currently the legal discussion about the compatibility of forensic data recorders and data protection regulations is still ongoing. Either a supreme court or legislature must decide about its legal admissibility. The proposed three implementation variants cover all plausible outcomes of this decision.

6 Conclusions

This paper proposed a method to store forensically valuable data within the telematic platform device in a secure and memory efficient manner. Utilizing the telematic platform allows direct access to a variety of data streams such as position data (GPS, cell site ID), passenger information (via Bluetooth/WiFi connected devices), vehicle status data (CAN/Ethernet messages, DTCs), etc. This data is highly beneficial for post-crime as well as post-crash related investigations. The proposed architecture implements a hybrid encryption scheme, using a symmetric and asymmetric cipher combination to achieve both privacy preserving, but forensically sound and tamper-proof storage of data.

Such an extensive ability to collect forensic information includes, however, almost always personal information of vehicle occupants. Based on the EU-GDPR, the manufacturer becomes the data processor for all personal data contained within the forensic storage, ensuing legal obligations for data protection. Due to the current unclear legal situation, three forensic storage variants were proposed, covering all possible future judicial decisions regarding this issue.

7 Outlook on Possible Online Architecture Extensions

This section discusses possible online extensions of the architecture proposal together with their trade-offs. They were not yet included into the proposal because of two reasons.

The first reason is the currently insufficient mobile broadband infrastructure in many countries. Additionally, the development and maintenance of

a server environment which could handle the data from millions of vehicles would be a huge expense for a company.

The most straightforward approach for an online extension is to directly upload all data to company servers. A poor connection could temporarily be mitigated with a volatile buffer. The advantage of this approach is that already sent forensic data is not affected by a compromised telematic platform. Nevertheless, forged data can be sent to the server once the compromise occurred. The problem of this approach is that a server infrastructure which can handle and securely store the forensic data of millions of vehicles is necessary. A centralized storage of the forensic data also means that there is a higher data privacy risk for information being leaked.

The above described backward tamper protection can also be accomplished without storing forensic information directly on a server. For this method only the authentication tags are uploaded with reference to the vehicle and associated data block. Although this approach does detect tampering and the deletion of previous data blocks, it cannot prevent it.

Finally, a method which does not require storing any data on the server is described. This is accomplished by letting the server sign a string containing the VIN (Vehicle Identification Number) and a timestamp for each session. The resulting signature is then appended to the plain text before encryption. That is $AE_{K_{sym}}(Compr(DB)_n \parallel TAG_{n-1} \parallel Sig(VIN + timestamp))$. Similar to the previous method, this procedure allows detecting tampering and deletion, but cannot prevent it.

In summary, online extensions provide backward tamper protection but cannot prevent tampering of new forensic data, as soon as a complete compromise of the device happened.

Acknowledgment

We would like to thank the reviewers for their constructive input which helped us to improve the overall architecture proposal.

References

[1] Nico Vinzenz and Tobias Eggendorfer. Forensic investigations in vehicle data stores. pages 1–6, 2019. ISBN 978-1-4503-7296-1. doi: 10.1145/3360664.3360665.

[2] H. Mansor, K. Markantonakis, R. N. Akram, K. Mayes, and I. Gurulian. Log your car: The non-invasive vehicle forensics. In *2016 IEEE Trustcom/BigDataSE/ISPA*, pages 974–982, 2016. doi: 10.1109/TrustCom.2016.0164.

[3] Tobias Hoppe, Sven Kuhlmann, Stefan Kiltz, and Jana Dittmann. It-forensic automotive investigations on the example of route reconstruction on automotive system and communication data. In Frank Ortmeier and Peter Daniel, editors, *Computer Safety, Reliability, and Security*, pages 125–136, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33678-2.

[4] National Marine Electronics Association (NMEA). Nmea 0183 datensätze, 2019. http://www.nmea.de/nmea0183datensaetze.html.

[5] European Commission. Digital single market – communication on exchanging and protecting personal data in a globalised world questions and answers, 2020a. https://ec.europa.eu/commission/presscorner/detail/en/MEMO_17_15.

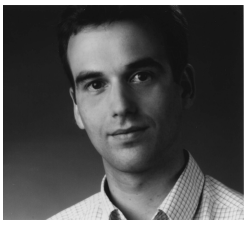
[6] European Commission. Adequacy decisions – how the eu determines if a non-eu country has an adequate level of data protection., 2020b. https://ec.europa.eu/info/law/law-topic/data-protection/international-dimension-data-protection/adequacy-decisions_en.

[7] SD Gleave, R FRISONI, F DIONORI, et al. Technical development and implementation of event data recording in the road safety policy. *Study for European Parliament*, 2014.

[8] David Hynd and Mike McCarthy. Study on the benefits resulting from the installation of event data recorders. *Study for European Commission*, 2014.

Biographies

Nico Vinzenz is Security Engineer at ZF Friedrichshafen AG, responsible for the security of autonomous mobility systems. Prior he studied IT security in Weingarten.



Tobias Eggendorfer is since 2013 professor of IT security in Weingarten, prior he was professor for IT-forensics in Hamburg since 2009.