Proposal for an achieveable, cost effective Security Concept for EOBRs

Revision 1.0

Continental
6755 Snowdrift Road
Allentown, PA 18106
+1 (610) 289-0488

Table of Contents

1. Introduction

Continental has significant experience in designing secure systems and building secure on-board recording devices.

The current definition of the EOBR system (395.16) does not specify a method for ensuring that the security level is sufficient to ensure that the system is acceptably tamper resistant. Systems that are designed to support the enforcement of restrictive legal requirements will always attract significant energy focused on circumventing those restrictions, especially when financial rewards are achievable.

2. Summary

This document contains a pragmatic proposal for an achievable level of security within the EOBR System. The proposal encompasses all elements and nodes of the System *by moving the focus of the security from secure transmissions to secure data*.

This concept has many benefits as it:

- is simple and easy to implement,
- is cost effective,
- fulfills all main security goals,
- is flexible and does not prescribe the communication technology for data transfer
- ensures enforcement can work with trustable, accurate data (no more comic books),
- reduces the ability of carriers to harass drivers

This concept allows a reliable access to the data for the enforcement and does not prescribe the communication technology to be used for data transfer. This will allow vendors to offer different systems adapted to the motor carriers requirements; will foster competition among vendors and technologies; will be future proof.

3. Problem

In the existing definition, the RODS files are unprotected, which means that several key security goals, which could be regarded as normal in such systems, cannot be achieved:

- Data integrity: needed to detect modifications to the data
- Authenticity: needed to guarantee the data is genuine (generated by a certified EOBR) and really belongs to a specific driver
- Non-repudiation: needed to avoid the driver being able to deny being the source of the data

These weaknesses seriously compromise the value of the RODS data to the enforcement community as the provided data cannot be fully trusted or can be repudiated. The ability to perform undetected modifications to the data also provides opportunities for the harassment of drivers.

Such modifications are not necessarily malicious, but could also be caused by hardware and software errors in systems which store and process the data. Even these accidental changes to the data must be detectable.

A very simple representation of the problem domain shows that the main weaknesses are at the nodes at which RODS data is stored. Even with state of the art secure communications between the system nodes, whenever the data is “at rest” at one of the nodes it is vulnerable to tampering/modification.

These weaknesses apply mostly to telematic based EOBR implementations due to the number of unprotected nodes involved. The risk of manipulations also exists in a peer to peer implementation, although it is significantly reduced, as the only weakness point is the EOBR itself.

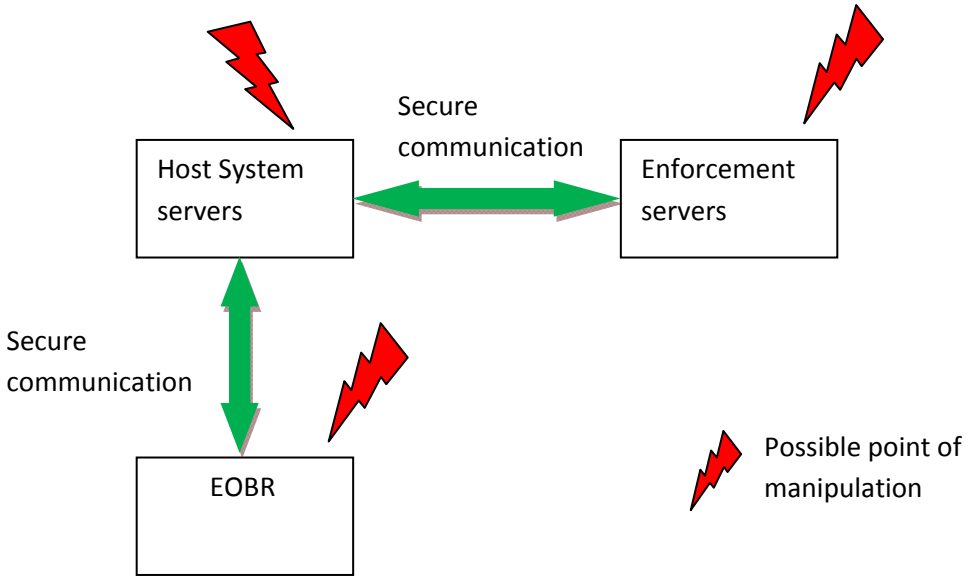

Figure 1: Problem domain

4. Solution

The following sections present a solution for achieving the missing security goals using digital signatures and public key infrastructure and encryption.

The focus of the solution is not to protect the complete path along which the data travels or could be stored, but to protect the data itself. Compared to reliably securing the complete data path, this is a highly robust, reliable and cost effective method to achieve the necessary protection.

The following concept description and non-exhaustive analysis presents various scenarios and demonstrates how the proposed concept addresses them. As this is a concept paper, not a specification, the level of detail is kept to a minimum.

4.1 Security Concept

4.1.1 General Architecture

The security concept is based on Public Key Infrastructure (PKI), in which every EOBR contains an asymmetric key pair, a secret key (SK) and a public key (PK). Each EOBR public key must be certified by a Certifying Authority (CA) who knows that the public key is genuine and belongs to the EOBR in question.

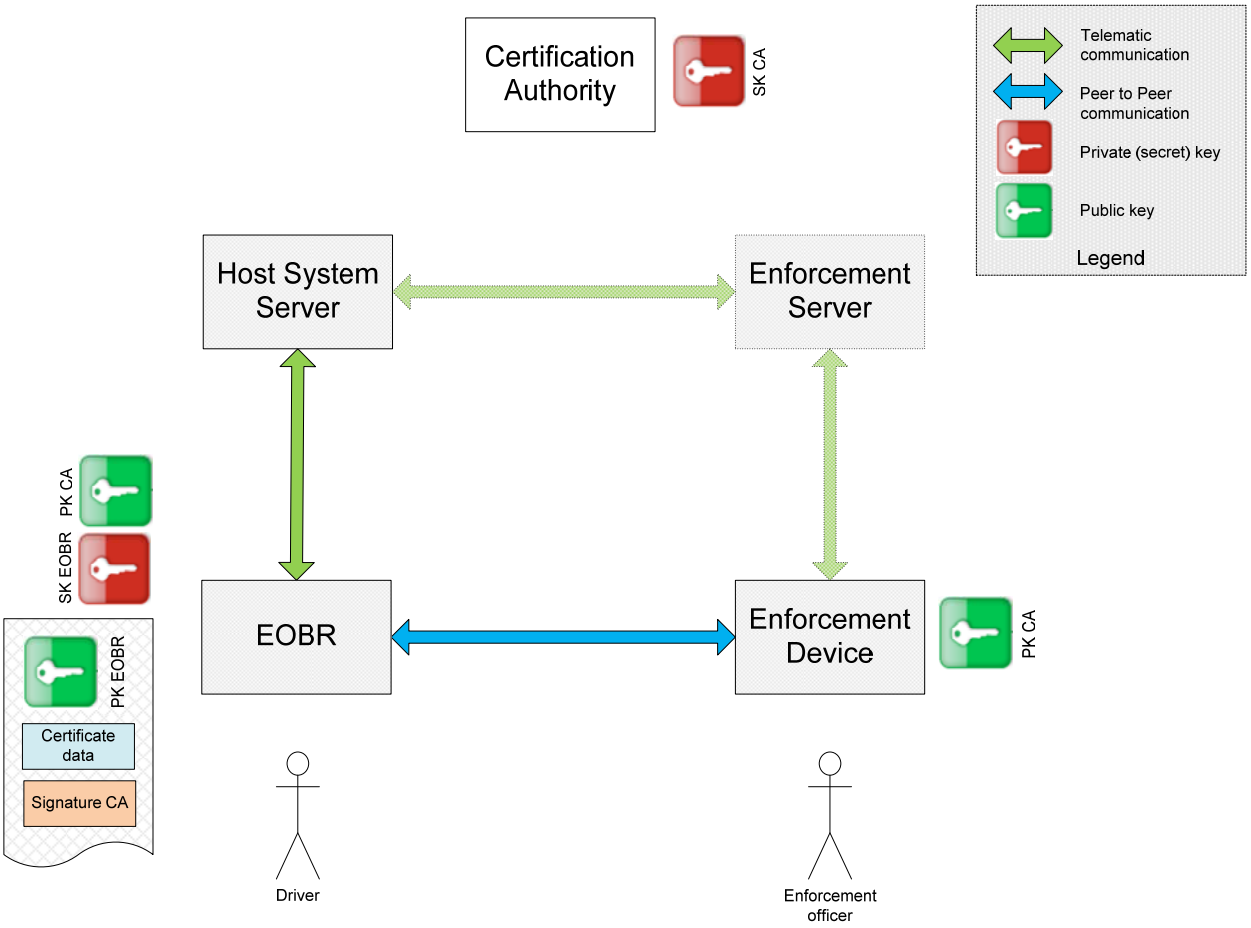

Figure 2: General Security Architecture

All relevant RODS data generated by the EOBR would have a digital signature appended which allows enforcement officers to ascertain the data authenticity and integrity and provides non-repudiation. For more details see 5.1.

4.1.2 Certification Authority (CA)

A CA is an entity that issues digital certificates. The digital certificate certifies the ownership of a public key by the named subject of the certificate. This allows enforcement officers to rely upon signatures made by the EOBR secret key that correspond to the public key that is certified.

The CA, its method of authorizing and its communication channels with the EOBR manufacturers should be examined by an independent security certification authority. The CA role may be performed by government or a trusted private enterprise.

The CA would also ensure that every EOBR that is issued with a certificate has a unique ID.

Such services could be provided by the Federal Technology Service (FTS) of the General Services Administration (GSA) through contracts under the Access Certificates for Electronic Services (ACES) program (http://aces.orc.com/).

4.1.3 Data Storage and Retrieval

395 Appendix A defines the data exchange format (RODS) as a flat file database. The addition of digital signatures should be based upon this definition. However in order to integrate the needed digital signatures and certificates and maintain the needed flexibility, some changes would be needed.

All EOBRs must store the data content of the signature relevant data elements internally in such a way that when the data is exported from the EOBR, the originally signed data can be reliably reassembled with the associated signature. Any change in the format or order of these elements would result in signature failure.

As long as the order is maintained, only the data values themselves need to be signed, not any meta data or tags that may be used to identify data in the RODS file.

4.1.3.1 Data Signature Sessions

A Data Signature Session (hereafter “Session”) represents a time period during which all relevant original data recorded in that period is protected by a single digital signature.

A Session could be started by starting a shift, driver login, driving or EOBR specific conditions (e.g. memory constraints).

A Session could be ended by the end of a shift, driver logout, day change, publishing of RODS data for enforcement purposes or EOBR specific conditions.

4.1.3.2 Data Elements

Several new data elements would be needed in addition to those already specified in the Data Elements Dictionary.

Here is a simplified view of potential new data elements:

Data Element	Data Element Definition	Type	Length	Sign
EOBR ID	Unique ID of the EOBR – same ID as in the certificate.	N	10 - TBD	Yes
Session Begin	Date and time of session begin	N	15	Yes
Session End	Date and time of session end	N	15	Yes
Digital Signature	ASCII representation of Digital Signature of all relevant data elements since the last Digital Signature was recorded	A	40-500	No
Certificate	ASCII representation of an EOBR public key certificate	A	130-1050	No
Record Signed	Indicates whether the record has been signed or not (optional depending on data storage concept – see 4.1.3.3)	A	1	

Table 1: Additional Data Elements needed

4.1.3.3 Data Storage

The data storage mechanism is so defined as to leave the maximum level of flexibility for the EOBR manufacturer, while providing the needed protection.

Every data element would have an additional attribute "Sign" in the data element dictionary. This attribute would indicate whether the data of that element, contained within an original data record, would be included in the signature calculation or not. Data that is not regarded as critical need not be included in the signature calculation. This would enable certain fields that are not regarded as security relevant to be omitted from the signature allowing them to be updated or edited.

There are two main alternative scenarios for organizing the data in an RODS file:

A) Original data and signature precedes annotated data

Such an RODS file could contain the following structure – only records colored blue are relevant for the signature calculation:

Session begin
Original Data Record 1 *(Event Update Status Code = C)*
Original Data Record 2 *(Event Update Status Code = H)*

Original Data Record 3 *(Event Update Status Code = C)*
Original Data Record 4 *(Event Update Status Code = H)*
Original Data Record 5 *(Event Update Status Code = C)*
Session end
Signature
Certificate
Annotated Data Record 2 (Event Update Status Code = C)
Annotated Data Record 4 (Event Update Status Code = C)
Session begin
……..etc.

This approach provides clear separation between the signed data and the unsigned data, and avoids the need to maintain an indicator as to whether or not a record has been signed as all records within the session are signed.

B) Original data and annotated data are interleaved

Such an RODS file would contain the following structure – only records colored blue are relevant for the signature calculation:

Session begin
Original Data Record 1 *(Event Update Status Code = C)*
Original Data Record 2 *(Event Update Status Code = H)*
Annotated Data Record 2 (Event Update Status Code = C)
Original Data Record 3 *(Event Update Status Code = C)*
Original Data Record 4 *(Event Update Status Code = H)*
Annotated Data Record 4 (Event Update Status Code = H)
Annotated Data Record 4 (Event Update Status Code = C)
Original Data Record 5 *(Event Update Status Code = C)*
Session end
Signature
Certificate
Session begin
……..etc.

In this case, it is no longer evident which of the records is original and which is annotated. Therefore each record would contain a new data element “Record Signed” indicating whether that record has been included in the signature calculation or not. This element indicates that the record to which it belongs is an original record or not.

This approach provides more flexibility in terms of data storage but requires a further data element to be stored per record.

4.1.3.4 Data Storage Rules

At least the following rules must be applied (non exhaustive list):

- whenever a new Session is started, a Session Begin is recorded
- whenever a Session is finished a Session End is recorded
- each Session must include an EOBR ID record
- the Digital Signature is calculated for all records between, and including, the Session begin and end records
- the Digital Signature is stored in the RODS file directly after the Session End
- the EOBR Certificate is stored in the RODS file directly after the Digital Signature
- the element Event Update Status Code must be excluded from the signature calculation (Sign = N)
- the EOBR must ensure that only, and all original records are signed, not annotations, regardless of the source of the annotation
- annotated records must not replace original records.

4.1.4 HOS calculations

HOS calculations are performed as today for carriers and EOBRs. The latest verified version of any record is used for the HOS calculation.

The EOBR will verify the signatures of all the RODS data, and indicate via the HMI whether the data is authentic or not. The EOBR will not use data that fails the signature verification in the HOS calculation.

An enforcement device could calculate and display both the HOS calculated from only the signed and verified data (where possible) and the HOS calculated using all data. Infringement information could be calculated for both versions. Significant discrepancies between the two values should be investigated. This approach could noticeably reduce the time required for a roadside control.

4.1.5 Key Generation and Installation

In order to maintain a system based on Public Key Infrastructure, it is necessary for each participating device to be fitted with a secret, or private key (SK) and a public key (PK) in the form of a certificate.

The following gives various examples how the keys could be generated and installed in the EOBR.

4.1.5.1 In a trusted factory

The asymmetric keys would be generated in a very trusted (certified) environment using non-EOBR hardware and/or software.

The PK and certificate data (serial number, company ID) are sent to the Certification Authority via a secure communication link, to create the EOBR certificate.

The certificate is returned to the factory and the SK and the certificate are installed into EOBR.

Disadvantage: SK is known to more than the EOBR alone. This increases the security requirement for the installation environment.

4.1.5.2 In a trusted factory or workshop

The asymmetric keys would be generated by the EOBR itself.

After exporting the PK from the EOBR it is sent, with the certificate data (serial number, company ID), to the Certification Authority via a secure communication link to create the EOBR certificate.

The certificate is returned to factory and installed into EOBR.

Advantage: the SK never leaves the EOBR which reduces the security requirements on the environment.

4.1.5.3 Over the Air SW Upgrade – EOBR Generates Keys

The Host System performs a SW upgrade over the air. This installs an EOBR software that is capable of generating, and protecting the needed key pair.

The next step requires a secure, private, mutually authenticated connection between the EOBR host system and a specific EOBR.

The EOBR generates the key pair, and sends only PK with the certificate data (serial number, company ID), via the secure connection, to the Host System.

The Host System obtains the certificate from the CA via a secure communication link and returns it to the EOBR.

Essential: SK never leaves the EOBR which makes this form of update possible.

4.1.6 Key Protection

The security of the system is scalable but the whole system is only as good as the weakest link. The EOBR SK must be protected in order for the system to remain trustable. A generalized view could define three different possible security levels for key protection and therefore the system as a whole (FIPS-PUB-199):

i. High: secure hardware will be needed inside the EOBR to protect SK from advanced attacks. This would mean that legacy systems might need to be extended if they cannot demonstrate such a level.

ii. Moderate: Keys and cryptographic algorithms can be partially protected by software in open systems. Various state-of-the-art techniques are available which can achieve a certain security level (e.g. http://www.whitecryption.com/).
A well-designed dedicated embedded system without specific secure hardware may be able to achieve this level as the effort to obtain the SK would need to be repeated for each individual system using specialist knowledge and tools.

iii. Low: SK stored unprotected in software in an open system. The Low level is not recommended for a recording unit.

It is important to note that even if a hacker succeeds in reading out the SK from one dedicated EOBR embedded system it will only allow to counterfeit RODS for this EOBR and not others. However, in an unprotected open platform with connection to the internet, the ability to install an Application that could retrieve the SK and allow the user to create rogue signed RODS data would seriously compromise the system, as the effort to hack one EOBR provides a solution that could directly apply to many EOBRS.

4.1.7 Certification

In order to ensure that the critical elements of the system:

- key generation and installation
- key storage
- EOBR generation, storage and processing of RODS records

are handled securely, correctly and consistently by all manufacturers, it is highly recommended that an EOBR certification system is implemented. This should be performed by a qualified independent body based on an agreed written specification (Security Target / Protection Profile) describing the security goals for all EOBRs according to an accepted security certification system e.g. Common Criteria.

An EOBR and its manufacturer must have achieved security certification before they are permitted to install any “hot” keys.

4.2 Roadside Enforcement

Whatever the data transfer approach used the implementation of a digital signature for the data allows enforcement officers to establish the validity of the data very quickly, and to exclude manipulation of the data. Repudiation of the data would also no longer be feasible.

As offline annotations are always visible for roadside enforcement, the motivation to perform such modifications reduces. This eradicates one of the sources of driver harassment.

The following paragraphs describe possible equipment and approaches that can be used by the enforcement for electronic roadside inspections. All data transfer approaches can be used, peer to peer and telematics.

The following main enforcement scenarios can be considered:

1. Full telematic solution as described in Issue 11 (MCSAC EOBR Sub-Committee document version 7.5.1 July 2011).
2. Partial telematic & peer to peer solution without an enforcement portal.
3. Full peer to peer solution.

More details of each scenario are described in chapter 4.3.

4.2.1 Full telematic solution

In this scenario the RODS data are transmitted via telematics from the EOBR and host system, via an enforcement portal to the enforcement device.

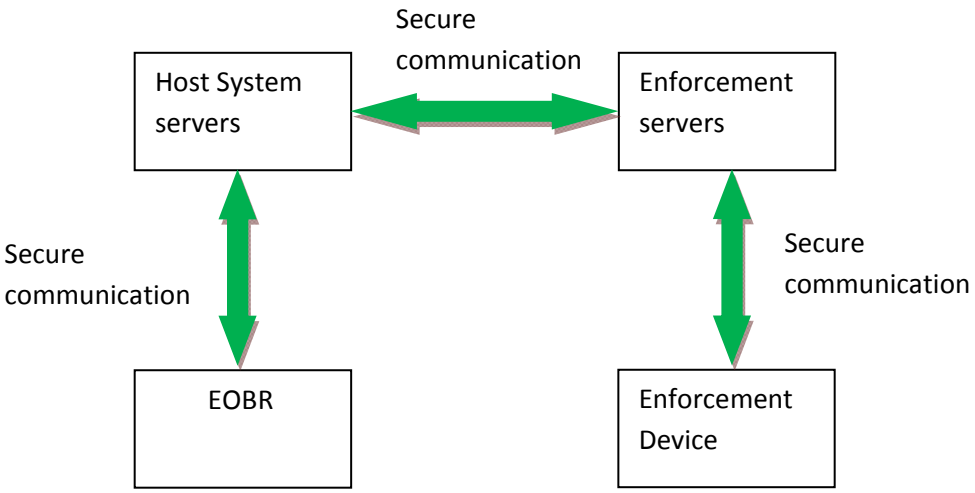

Figure 3: Nodes in a full telematic inspection

If no network capability is available when the roadside inspection is being carried out, the inspection may not be possible.

4.2.2 Partial telematic & peer to peer solution without an enforcement portal

The following scenario allows for the use of telematics on the EOBR side and removes the requirement for:

- enforcement devices or vehicles to have network connectivity and for
- the enforcement community to maintain an enforcement portal and servers.

When a roadside inspection is requested, the EOBR would check whether it has all necessary data to provide a full RODS, if not, it would request the data from the host system. Rather than supplying the data to an enforcement system, the host system would send the data back to the EOBR. The EOBR then supplies the data to the enforcement device. Due to the fact that the data is signed, the integrity is guaranteed. Even if the EOBR does not have network connectivity during a particular roadside inspection, the EOBR can provide all the data stored in its memory.

This approach would make the telematic approach required by fleet management systems totally transparent to the enforcement community.

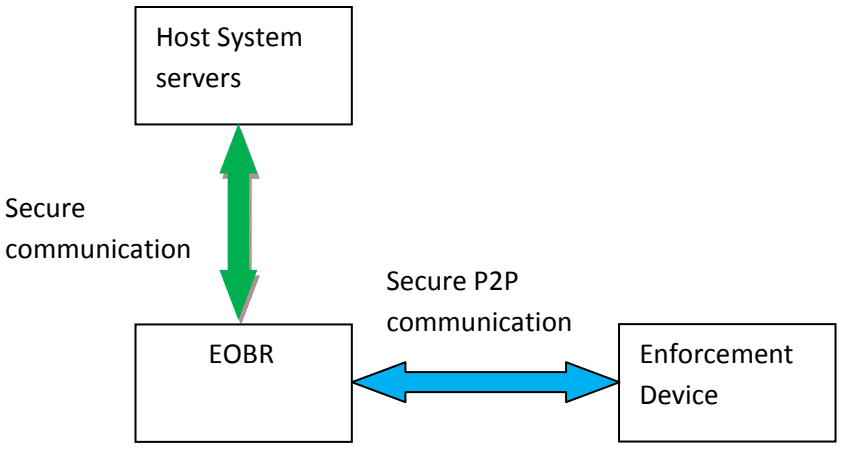

Figure 4: Nodes in a mixed telematic / peer to peer inspection

4.2.3 Full peer to peer solution

In this scenario the RODS data are all available in the EOBR and are transmitted to the enforcement device via a peer to peer connection. No network capability is needed for the EOBR or the enforcement device.

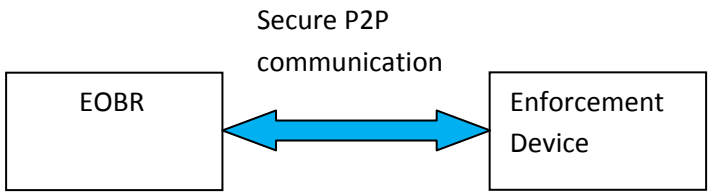

Figure 5: Nodes in a peer to peer inspection

4.2.4 Enforcement Equipment

4.2.4.1 Dedicated Roadside Enforcement Device

It is recommended that roadside enforcement officers be equipped with a dedicated mobile enforcement device that can acquire data either via peer to peer or via telematic connection if needed (see 4.2.4.2). The device should provide instant verification and full HOS analysis of the data as well as storing acquired data until the next back-up opportunity. Results of the HOS analysis and details of any infringement would be provided on a built-in display.

Such a device should be:

a) capable of accepting a mobile storage device, such as a USB stick , to transport the RODS data and

b) equipped with a bluetooth wireless interface.

As the USB stick would only be inserted into the dedicated enforcement device, there would be no danger to standard enforcement systems. The dedicated enforcement device would connect to the enforcement system via an accepted connection e.g. WLAN, Bluetooth, LAN, RS232 etc. The USB stick solution would only need to function as a mass storage device.

For EOBRs without a USB interface, a bluetooth connection would be used. An exchange of a pairing PIN would be needed for authentication. For more details see the document “Approach for Secure Peer to Peer Roadside Log Downloads for EOBRs” distributed at the MCSAC Meeting on 1-2nd August 2011.

Such a dedicated solution would be robust, designed to exactly fulfill its purpose, cost effective and easy to use.

4.2.4.2 Standard Enforcement Computer with network connectivity

A standard enforcement computer with network connectivity can perform data transfer either via peer to peer communication using Bluetooth or USB (for the states that allow to use such interfaces) or via telematics.

4.2.4.3 Roadside Enforcement computer without network connectivity

A standard enforcement computer without network connectivity can perform data transfer via peer to peer communication using Bluetooth or USB (for the states that allow the use of such interfaces).

4.3 Use Cases

The following Use Cases are outlined to demonstrate the handling of the concept in everyday use:

A) Uploading/archiving of RODS data to a host system via telematics
B) Archiving of RODS data via Mobile Storage Device
C) Roadside Enforcement using a full telematic solution
D) Roadside Enforcement using a partial telematic & P2P solution
E) Roadside Enforcement using an electronic peer to peer solution
F) Roadside Enforcement using a printer
G) Annotation of RODS records by the carrier
H) Annotation or modification of RODS records by the driver
I) Driver moves from truck to truck within one carrier
J) Driver moves from truck to truck between different carriers

4.3.1 UC_A: Uploading/archiving of RODS data to a host system via telematics

Context of use: Regular uploads of RODS data to carrier

Main success scenario:

1. RODS data from completed Sessions will be uploaded with their signatures/certificates.
2. RODS data from an ongoing Session will be uploaded without signature/certificate.
3. At the end of a Session, the RODS data must be uploaded to the host system again, to ensure the signed records with their signatures are available.

Note: the EOBR must maintain all original records of an open session in its memory. At the end of a Session, the EOBR signs the original records, not the annotated records.

4.3.2 UC_B: Archiving of RODS data to a Mobile Storage Device

Context of use: Storage of RODS data to a Mobile Storage Device.

Pre-condition: Driver logged in to EOBR with a Mobile Storage Device.

Success End Condition: RODS data stored on a Mobile Storage Device with signatures/certificates.

Trigger: EOBR has detected the end of a Session.

Main success scenario:

1. A Session ends.

2. The EOBR closes the Session.
3. The EOBR calculates the signature for the completed Session.
4. The EOBR stores the Session data on the Mobile Storage Device with the signature/certificate.

4.3.3 UC_C: Roadside Enforcement using a telematic solution

Context of use: An enforcement officer wants to perform a roadside inspection

Pre-condition: Telematic infrastructure available, driver is logged in.

Success End Condition: Enforcement officer receives the requested RODS data and can verify its validity.

Trigger: Enforcement officer requests the RODS data for a roadside inspection.

Main success scenario:

1. Enforcement Officer requests the RODS data for a roadside inspection.
2. Driver triggers roadside inspection upload process.
3. The EOBR closes the Session and calculates the signature.
4. The EOBR uploads the RODS data including the signatures/certificates to the host system.
5. The host system puts the data to the enforcement system.
6. The host system returns the Identification Key to the EOBR
7. The driver provides the enforcement officer with the Identification Key
8. The enforcement officer retrieves the RODS data from the enforcement server using the Identification Key.
9. The enforcement officer validates the data using the digital signature.
10. The enforcement officer performs the inspection using an enforcement device and the validated data.

4.3.4 UC_D: Roadside Enforcement using a partial telematic & P2P solution

Context of use: An enforcement officer wants to perform a roadside inspection

Pre-condition: EOBR with an electronic Peer to Peer solution and host system is available

Success End Condition: Enforcement officer has performed roadside inspection

Trigger: Enforcement officer requests the RODS data for a roadside inspection

Main success scenario:

1. Enforcement Officer requests the RODS data for a roadside inspection.
2. Driver triggers roadside inspection process via EOBR user interface.

3. If the EOBR does not have all the data stored locally, it obtains the data from its host system.
4. The RODS data is transferred from the EOBR to an enforcement device via a Peer to Peer connection e.g. Bluetooth or USB stick.
5. The enforcement device validates the data using the digital signature and performs the HOS and infringement analysis.
6. The enforcement officer performs the roadside inspection using the validated data from the enforcement device.

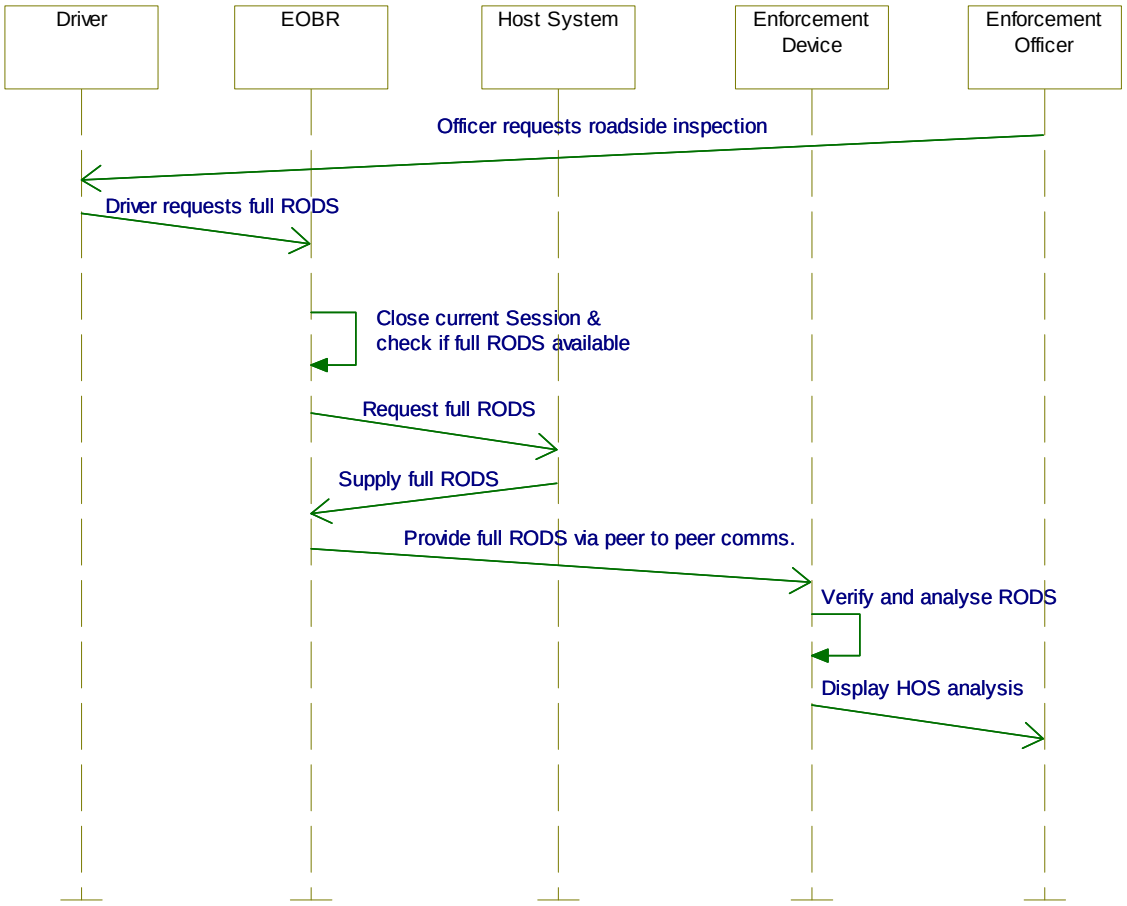

Figure 6: Simplified view of a roadside enforcement with no enforcement portal

4.3.5 UC_E: Roadside Enforcement using an electronic peer to peer solution

Context of use: An enforcement officer wants to perform a roadside inspection

Pre-condition: EOBR with an electronic Peer to Peer solution is available

Success End Condition: Enforcement officer has performed roadside inspection

Trigger: Enforcement officer requests the RODS data for a roadside inspection

Main success scenario:

1. Enforcement Officer requests the RODS data for a roadside inspection.
2. Driver triggers roadside inspection process via EOBR user interface.
3. The RODS data is transferred from the EOBR to an enforcement device via a Peer to Peer connection e.g. Bluetooth or USB stick.
4. The enforcement device validates the data using the digital signature and performs the HOS and infringement analysis.
5. The enforcement officer performs the roadside inspection using the validated data from the enforcement device.

4.3.6 UC_F: Roadside Enforcement using a printer

Context of use: An enforcement officer wants to perform a roadside inspection

Pre-condition: EOBR with printer is available

Success End Condition: Enforcement officer has performed roadside inspection

Trigger: Enforcement office requests the RODS data for a roadside inspection

Main success scenario:

1. Enforcement Officer requests the RODS data for a roadside inspection.
2. The driver triggers a roadside inspection printout.
3. The EOBR closes the Session and calculates the signature.
4. The EOBR verifies the signatures of the complete available RODS history.
5. The EOBR prints out the RODS, and the driver provides it to the enforcement officer.
6. The enforcement officer performs the inspection.
7. In the event that the printout needs to be retained as evidence, the enforcement officer requests that the driver authenticate the printout by signing it.
8. The driver signs the printout.

4.3.7 UC_G: Annotation of RODS records by the carrier

Context of use: The carrier wants to annotate RODS data

Pre-condition: RODS file available with records that need annotating

Success End Condition: Annotated records are stored in the RODS file

Main success scenario:

1. The carrier wants to annotate an existing RODS file.
2. The carrier performs the annotation using the appropriate software. No signature relevant data elements in the original records may be modified, but a new record containing the annotation is created and edited.
3. The “Event Update Status Code” element in the original record is changed to “H” for Historical.
4. The “Event Update Status Code” of the annotated record is set to “C” for Current.
5. The records are saved in the RODS file by the carrier.
6. The changes are transferred back to the EOBR, if needed.
7. The EOBR stores the changes, without modifying any original signature relevant data elements.

4.3.8 UC_H: Annotation or modification of RODS records by the driver

Context of use: The driver wants to annotate RODS data

Pre-condition: Driver is logged into the EOBR and a session is open. The EOBR supports driver editing of non-driving driver status records.

Success End Condition: Annotated or modified records are stored in the RODS file

Trigger: Driver annotates or modifies records in an RODS file

Main success scenario:

1. The driver annotates an existing record using the EOBR, if the EOBR supports this feature.
2. No signature relevant data elements in the original records are modified, but the EOBR creates a new record containing the annotation.
3. The “Event Update Status Code” of the original record is changed to “H” for Historical.
4. The “Event Update Status Code” of the annotated record is set to “C” for Current.
5. The records are saved in the RODS file.

4.3.9 UC_I: Driver moves from truck to truck with one carrier

Context of use: A driver works in different vehicles with one carrier and host system (telematic)

Pre-condition: Both vehicles fitted with an EOBR and either telematics or portable storage medium (e.g. USB stick)

Success End Condition: Complete HOS / RODS records available

Trigger: Truck Change

Main success scenario:

1. The driver logs out of EOBR in Truck 1.
2. The session is completed and the RODS stored – either via upload to host system or on portable storage medium.
3. The driver logs into Truck 2.
4. Data downloaded from host system or available from portable storage medium.
5. Complete RODS records available for HOS calculations.
6. EOBR performs HOS calculation using data that has passed the signature verification.

4.3.10 UC_J: Driver moves from truck to truck with different carriers

Context of use: A driver works in different vehicles with belonging to different carriers

Pre-condition: Both vehicles fitted with an EOBR and either telematics or portable storage medium. The data storage for each carrier is separate.

Success End Condition: Complete HOS / RODS records available

Trigger: Truck Change

Main success scenario:

1. The driver logs out of EOBR in Truck 1.
2. The session is completed and the RODS stored – either via upload to host system or on portable storage medium.
3. The driver enters Truck 2 and logs in.
4. Authenticated data from previous vehicle only available when using portable storage medium.
5. Complete, authenticated RODS records available for HOS calculations when using portable storage medium.
6. EOBR performs HOS calculation using any available data that has passed the signature verification.

Note: this scenario highlights that without a portable storage medium that the driver is mandated to carry (electronic log book) or central data storage (privacy issue), it is technically still possible to seriously exceed the HOS.

5. Appendix

5.1 Digital Signatures

Digital Signatures are used to provide authenticity, integrity of data and non-repudiation. The signature consists of a hash value that has been encrypted using a secret key from a key pair that is being used to implement public-key cryptography.

There are various implementations of public-key cryptography e.g. RSA and elliptical curves. Using elliptical curve cryptography reduces the size of the signatures and the certificates.

A typical signature length using RSA would be 128-256 byte and using elliptical curves 20-28 byte.

5.1.1 Creating a Digital Signature

In order to ensure that any modification of the data to be protected can be detected, two steps are needed:

a) *Create a hash using a public algorithm e.g. MD5, SHA1, SHA2*

Figure 7: Create a hash “fingerprint” of data

If a suitable algorithm is used, the hash represents a unique “fingerprint” of the data to be protected.

b) *Encrypt the hash using asymmetrical encryption with the EOBR secret key (SK)*

To create the signature, the generated hash is encrypted using the EOBR SK. The result is a package consisting of the Data and the Digital Signature. These two elements need to be stored together.

5.1.2 Verifying a Digital Signature

In order to verify a digital signature and thereby verify the source and integrity of the data, the following elements are needed:

- The public key (PK) of the Certification Authority (CA), PK CA

- The certificate of the EOBR from which the data originated
- The data packet to be verified with its digital signature

The following steps are then performed:

a) *Validate the EOBR public key certificate:*
   - decrypt the signature using the CA PK to get the hash in the certificate
   - calculate the hash of the certificate itself
   - compare the calculated version of the hash with the decrypted version. If they match the certificate is genuine.
   - Additionally the validity time frame of the certificate can be checked from the certificate data.

   Once this test has been passed, it is certain that a valid public key has been provided with which to verify the EOBR data.

b) *Check the signature of the EOBR data:*
   - use the public key from the certificate to decrypt the signature using the EOBR PK to get the hash from the signature in the data packet
   - calculate the hash of the data packet itself
   - compare the calculated version of the hash with the decrypted version. If they match the data must be from the EOBR that created it, and has not been modified.

Provided the trust anchor of the Certificate Authority can be relied upon, this is a simple and reliable method to verify the data authenticity and integrity.

5.2 Glossary

5.2.1 Integrity

Integrity is the property that data is protected against inadvertent or unauthorized modification.

5.2.2 Authenticity

Authenticity is the property that data is genuine and originated from its purported source

5.2.3 Authentication

Authentication is the process of establishing confidence in user identities.

5.2.4 Repudiation

Claiming to have not performed an action. Repudiation occurs when someone performs an action and then claims that they didn't actually do it.

5.2.5 Confidentiality

The prevention of disclosure of information to unauthorized individuals or systems.

5.2.6 Original Data

Any data recorded directly by an EOBR, without any editing or annotation.

5.2.7 Annotated data

Any data that was not directly recorded by an EOBR, but has been edited, updated or added either using the EOBR or in a back office environment.

Manually entered driver status records are regarded as annotations and would not be signed TBD.

5.2.8 Common Criteria

[Wikipedia] Common Criteria is a framework in which computer system users can *specify* their security *functional* and *assurance* requirements, vendors can then *implement* and/or make claims about the security attributes of their products, and testing laboratories can *evaluate* the products to determine if they actually meet the claims. In other words, Common Criteria provides assurance that the process of specification, implementation and evaluation of a computer security product has been conducted in a rigorous and standard manner.

CC originated out of three standards:

- ITSEC - The European standard, developed in the early 1990s by France, Germany, the Netherlands and the UK. It too was a unification of earlier work, such as the two UK approaches (the CESG UK Evaluation Scheme aimed at the defense/intelligence market and the DTI Green Book aimed at commercial use), and was adopted by some other countries, e.g. Australia.

- CTCPEC - The Canadian standard followed from the US DoD standard, but avoided several problems and was used jointly by evaluators from both the U.S. and Canada. The CTCPEC standard was first published in May 1993.

- TCSEC - The United States Department of Defense DoD 5200.28 Std, called the Orange Book and parts of the Rainbow Series. The Orange Book originated from Computer Security work including the Ware Report, done by the National Security Agency and the National Bureau of Standards (the NBS eventually became NIST) in the late 1970s and early 1980s. The central thesis of the Orange Book follows from the work done by Dave Bell and Len LaPadula for a set of protection mechanisms.

CC was produced by unifying these pre-existing standards, predominantly so that companies selling computer products for the government market would only need to have them evaluated against one set of standards. The CC was developed by the governments of Canada, France, Germany, the Netherlands, the UK, and the U.S.

5.2.9 Elliptic curve cryptography

[Wikipedia] It is believed that the same level of security afforded by an RSA-based system with a large modulus can be achieved with a much smaller elliptic curve group. Using a small group reduces storage and transmission requirements.

The U.S. National Security Agency has endorsed ECC by including schemes based on it in its Suite B set of recommended algorithms and allows their use for protecting information classified up to top secret with 384-bit keys.[3] While the RSA patent expired in 2000, there are patents in force covering certain aspects of ECC technology, though some argue that the Federal elliptic curve digital signature standard (ECDSA; NIST FIPS 186-3) and certain practical ECC-based key exchange schemes (including ECDH) can be implemented without infringing them.