Conference Paper

Assessment of Threats to the Security of the Cryptographic Authentication Mechanisms of the Monitor Devices of Vehicles

Victor S. Gorbatov[1], Igor Yu. Zhukov[2], and Oleg N. Murashov[3]

[1]National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Kashirskoe shosse 31, Moscow, 115409, Russia
[2]CEO of Ltd. «The National Mobile Portal», Volgogradskiy pr.,2 off.36, Moscow, 109316, Russia
[3]Joint-stock company «Ramec-VS», Volgogradskiy pr., 2, Moscow, 109316, Russia

Abstract

In accordance with the legislation on transport security, a number of vehicles must be equipped with on-Board control devices containing a cryptographic means of authentication, registration and storage of control data, including key information of the electronic signature.

This paper presents a solution to the problem of justification of the adequacy of measures to counter known attacks and methods of discrediting the suggested cryptographic mechanisms and the corresponding protocol, drawn up in the form of a draft national standard and presented in the previous work of the authors devoted to study of its security properties. The solution presented is limited to the consideration of attacks divided into two large classes: passive and active attacks, including temporary attacks based on the study of the response time of one or more participants of the protocol.

The analysis of the security threat model of the Protocol generating a common key with the authentication of subscribers intended for use in tachographs installed on vehicles shows that the protocol provides sufficient measures to counter known attacks. The found possible attacks are of a formal nature, not allowing the offender to obtain any additional information in order to discredit the protocol.

Corresponding Author:
Victor S. Gorbatov
VSGorbatov@mephi.ru

Received: 22 July 2018
Accepted: 9 September 2018
Published: 8 October 2018

Publishing services provided by Knowledge E

Selection and Peer-review under the responsibility of the Breakthrough Directions of Scientific Research at MEPhI Conference Committee.

1. Introduction

In accordance with the current rules of transport security [1, 2] a certain category of vehicles must be equipped with on-Board control devices containing a cryptographic means of authentication, registration and storage of control data, including key information of the electronic signature. Cryptographic mechanisms and related protocol developed in accordance with the recommendations [3] and designed as a draft of the

national standard [4], as well as its security properties are presented in [5]. The problem of substantiation of sufficiency of measures of counteraction to the known attacks and methods aimed to discredit the specified protocol for the purpose of development of recommendations for its practical application was formulated in [5] as well. The present paper is devoted to the solution of this problem.

The obvious step in achieving a positive solution to the problem is the development of a threat model (possible types of attacks) and the analysis of appropriate measures to counter these threats.

Since in this paper we consider only the model of the cryptographic protocol, and not its hardware-software implementation for a particular technical means of cryptographic protection, it is advisable to limit the consideration of only attacks divided into two large classes: passive and active attacks, including temporary attacks based on the study of the response time of one or more participants in the protocol.

2. Passive Attacks

Passive attacks are based on perlustration and subsequent cryptographic analysis of messages transmitted during the protocol execution. Therefore, let us assume that before the protocol starts, the intruder has some information about a certain number of on-board devices $VU_1, ldots, VU_v$ and tachograph cards $TC_1, ldots, TC_t$, where t, v are natural numbers and $v \geq 1, t \geq 1$, for which known are:

1. The identifiers of the protocol parties, respectively:

$$VU_1.CHR, \ldots, VU_v.CHR \text{ и } TC_1.CHR, \ldots, TC_t.CHR;$$

2. Keys for verification of electronic signature, respectively:

$$VU_1.P, \ldots, VU_v.P \text{ и } TC_1.P, \ldots, TC_t.P;$$

3. Certificates of electronic signature verification, keys signed by the electronic signature of the Certifying center and containing both the values of the electronic signature verification keys and the values of the identifiers, respectively *VU.CHR* и *TC.CHR*.

In addition, the violator is aware of the agreed in advance parameters *a, b, p* of the elliptic curve $E_{a,b}$ and the point $P \in E_{a,b}$, which generates a subgroup of prime order *q*. Further, we assume that these parameters are the same for all possible sessions of the protocol. In addition, the violator is aware of all cryptographic algorithms: signature

generation and verification, information encryption algorithms, as well as the function used to calculate the derived key. Thus, in accordance with the scheme of the protocol under study [4], during one session of the protocol execution, the following values become known to the offender.

1. Points at the elliptic curve *TC.P*, *VU.P*, which are used to generate a shared session key.
2. Random sequence $Nonce_1$, transmitted from the tachograph card to the on-board device.
3. Ciphertext E_1, being the result of the encryption algorithm GOST R 34.12-2015 «Magma» working in XOR cipher regime with respect to unknown block $Nonce_2$ 64 bit long while an to offender key K and an unknown to offender initialization vector (synchrosignal I).
4. The value of electronic signatures S_1, S_2, calculated for unknown to offender messages T_1, T_3 with the help of unknown to the offender secret long-term keys (electronic signature keys *TC.SK*, *VU.SK*).

Using the above values, the offender can implement the following threats aimed to discredit the protocol.

2.1. Attack on long-term keys

Since the offender is aware of the agreed in advance parameters *a, b, p* of the elliptic curve $E_{a,b}$ and the point $P \in E_{a,b}$, which generates a subgroup of prime order *q*, the task of determining the long-term key *VU.SK* of the on-board device is reduced to the solution of the problem of discrete logarithm *VU.PK = [VU.SK]P* in the elliptic curve points group $E_{a,b}$. Solving a similar problem *TC.PK = [TC.SK]P* will allow the offender to find the long-term key for the tachograph card.

It is known that at present the best method for solving the problem of discrete logarithm in the group of points of an elliptic curve is the method of parallel search of Oorshot-Wiener collisions [6].

The labor intensity of this method is estimated by $\frac{\pi q}{2}$.

Here and further, the labor intensity is measured in the operations of adding different points of an elliptic curve $E_{a,b}$, where *q* is the order of the subgroup generated by the point *P*, the factor 2 in the denominator of the reduced ratio means the number of effectively computable automorphisms of the elliptic curve. Taking into account that

according to the GOST R 34.10-2012 for q the following inequalities $2^{254} < q < 2^{256}$ are valid, one can assume that the labor intensity of the solution of the problem of discrete logarithm is estimated by the order of magnitude as 2^{128}.

2.2. Attack via solution the Diffie-Hellman problem

Let us assume $G =< P >$ is a subgroup of elliptic curve points $E_{a,b}$ of prime order q, generated by point P. Assume this subgroup have two elements $R_a = [k_a]P$ и $R_b = [k_b]P$. We will call the Diffie-Hellman problem the problem of finding the element Q, satisfying the equality $Q = [k_a k_b]P$. In the case of the protocol we are considering a role of the point R_a plays transmitted during the protocol execution point $VU.P$, and a role of the point R_b plays the point $TC.P$. Then, the solution of the Diffie-Hellman problem will be a point $VU.Q$, calculated by the on-board device at the third step (p.4) and a point $TC.Q$, calculated by the tachograph card at the fourth step (p.1.2) of the above protocol. It is easy to see that the solution of the Diffie-Hellman problem by the offender leads to the determination of an unknown common session key K.

Currently, only one effective method for solving the Diffie-Hellman problem is known, which is different from the total enumeration of unknown values. This method was discussed above and is based on the solution of the discrete logarithm problem in the group of points of an elliptic curve $E_{a,b}$.

Thus, we can assume that the labor intensity of the Diffie-Hellman problem coincides with the labor intensity of the solution of the discrete logarithm problem and is also estimated by the order of magnitude 2^{128}.

2.3. Finding a common key K with ciphertext E_1

The problem of finding the common session key K with ciphertext E_1 is reduced to determining the value of K, satisfying the following system of nonlinear equations:

$$E_1 = Nonce_2 \oplus E(K, I)$$

$$K||I = [KDF(\pi(VU.Q)|\ |VU.CHR|\ |TC.CHR)]_{0,\ldots,319},$$

with the unknown values $Nonce_2 \in V^{64}$ and $\pi(VU.Q) = \pi(TC.Q) \in V^{256}$.

Methods for solving this system of equations, different from the total testing of values $\pi(VU.Q)$ can be estimated as large as $O(2^{256})$ *of* mathematical operations.

2.4. Attack on a one-time key of the algorithm of electronic signature generation

By the one-time key of the algorithm for generating an electronic signature, we mean a random number *k*, produced in the process of calculating the electronic signature and used in the equation for generating the signature.

$$rx + ke \equiv s(\mathrm{mod} q).$$

In the analysis of the electronic signature scheme GOST R 34.10-2012 it is considered that the values *r, s, e* are known to the offender (pair *r* // *s* is the value of the electronic signature, and the value *e* is the value of the hash function of the signed message).

In the case of the Protocol under study, the unknown *x* can take the following values *VU.SK* or *TC.SK*, therefore it is a long-term key that is not available to the offender. The value of e is also not known to the offending party because the signed message T_1 or T_3 is fully unknown to te offender. Thus, if the offender knows the value of the one-time key k, then he needs to try 2^{64} unknown values $Nonce_1$ or $Nonce_2$, and for each of these values to calculate the value of *e* and solve the specified linear equation.

2.5. Attack on one-time random values

While executing the protocol by the on-board device and the tachograph card the following random integers k_t, k_b are calculated that satisfy the inequalities:

$$1 \leq k_t k_b \leq q - 1$$

It is easy to see that if the offender knows at least one of these values, he can determine the common session key.

Indeed, if the offender knows the value of k_t, generated by the tachograph card, then by intercepting the point *VU.P* the offender can compute the point *Q* and the common secret key using the following equality:

$$I = [KDF(\pi(VU.Q)| \ |VU.CHR| \ |TC.CHR)]_{0,\ldots,319}, \quad \text{where } Q = [K_t]VU.P.$$

2.6. Attack on the pseudorandom numbers generator

Another possibility to discredit the protocol is an attempt to predict the pseudo-random number used by the on-board device or tachograph card and generated during the protocol execution.

During one session of the protocol execution the offender has the opportunity to observe the $Nonce_1$ sequence of a fixed length of 64 bits produced by the tachograph card. Moreover, the offender can observe this sequence for a sufficiently large number of sessions of the protocol execution and accumulate the collected data.

We believe that the used random number generators produce sequences of uniformly distributed unpredictable 64-bit integers. In this case, given the set of numbers, it is impossible to determine the sequence of numbers generated earlier or that will be generated later in real time. General requirements for such generators can be found in the recommendations for standardization of cryptographic protocols [3].

2.7. KCI and UKS attacks

A detailed analysis of these types of threats and recommendations for combating them is given in [5].

3. Formal Analysis of Active Attacks

One of the possible ways to study the protocols on the possibility of active attacks is the mathematical modeling of the actions of the offender with the help of a wide range of means of automatic verification of cryptographic protocols. We used available and well-studied means of automatic verification of cryptographic protocols *AVISPA – SPAN* [5, 9] and *Scyther* [7].

3.1. Analysis using the tool AVISPA-SPAN

Modeling of the protocol under study was carried out in accordance with the specification, see [3].

The analysis using the *AVISPA – SPAN* automatic verification tool [8, 9] was performed using *OFMC* and *CL – AtSe* modules that perform verification by the model verification method. The description of the modules used is given in [10]. We study the security level by means of built-in *AVISPA – SPAN* tools functions: *Secret*, *Witness* and *Request*. The function *Secret* validates the security and applies to any data involved in the protocol execution process. In particular, it can be applied to the study of the security of the generated public key, that is, to the verification of the first security property [5]. In our study the Secret function was also applied to one-time secret keys k_t, k_b — secret random values generated during the protocol execution. The functions

Witness и Request allow to check the possibility of secure authentication at a certain (specific) step of the protocol execution. Thus, the use of the *Witness* and *Request* functions allows to check the sixth security property [5].

The results of the analysis of the protocol under study [4] using *AVISPA – SPAN* showed that the cryptographic mechanism under consideration allows a secure authentication and the generation of a common secret key. The results of the analysis are shown in the following table.

Table 1

Module	Details	Property	Result
OFMC	*BOUNDED_NUMBER_OF_SESSIONS*	Authentication	*SAFE*
OFMC	*BOUNDED_NUMBER_OF_SESSIONS*	Secrecy	*SAFE*
CL-ATSE	*BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL*	Authentication	*SAFE*
CL-ATSE	*BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL*	Secrecy	*SAFE*

The scheme of execution of the model of the cryptographic mechanism under study with *AVISPA – SPAN* is shown in Fig 1.

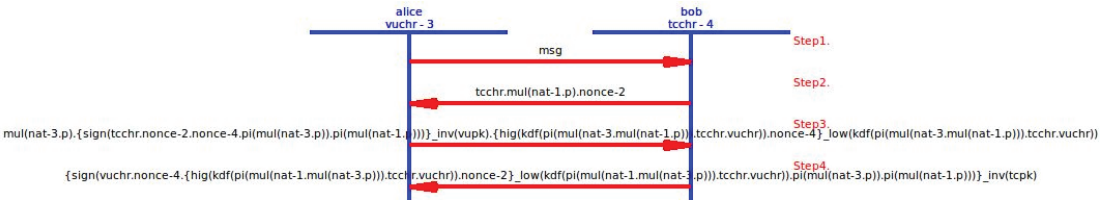

Figure 1: Protocol evaluation with *AVISPA–SPAN*.

3.2. Analysis using Scyther tool

The *Scyther* tool uses the *SPDL* specification language, which allows to define a set of states and an interstate transition system. While investigating the protocol, symbolic analysis is used in combination with bidirectional search based on partially ordered patterns [10, 11]. The formal protocol model used in *Scyther* describes a set of states and a system of transitions from one state to another. States that are reachable from a given initial state are checked to satisfy some security properties. The protocol is defined as a sequence of events. The events include both the transmission of messages exchanged between the protocol participants and the messages that an attacker can send. *Scyther* verifies limited and unlimited number of protocol sessions. A notation is used to distinguish between individual events. *Scyther* does not require to knowing

the attack scenario beforehand. One only needs to set parameters that limit either the maximum number of starts or the trajectory space.

When verifying the protocol, *Scyther* considers a number of security claims first proposed in [12] and chosen by Scyther developer to justify the security of the investigated models of cryptographic protocols.

The following security claims were used in the analysis of the protocol under study: *Secret*, *Alive*, *Weakagree*, *Niagree и Nisynch*.

The *Secret* claim allows to verifying the secrecy property performance, which is similar to the property used by AVISPA – SPAN. Definitions of other claims can be found in [12, 13].

These security properties do not exactly match the properties we have introduced in [5] and have their own interpretation. In this regard, we present the definitions of these properties in the interpretation closest to those considered in [5].

1. When the *Alive* claim is satisfied, the protocol guarantees to the participant the aliveness of the other participant, provided the participant acting as the initiator of the protocol, after the protocol is completed, as he believes, with the other participant, receives confirmation that he was actually a participant of this protocol. Note that it is not necessary for a participant to think that he or she is interacting with the participant, and that the protocol was started by the participant before the protocol was completed by the participant. If for the *Alive* claim to demand that the participant believes that he or she interacts with the participant, then we get the claim definition *Weakagree*.

2. When the claim *Weakagree* is satisfied, the protocol guarantees to the participant A *weak agreement* with the other participant, provided the participant acting as the initiator of the protocol, after the protocol is completed, as he believes, with the other participant:

    - receives confirmation that he was actually a participant of this protocol;
    - believes that he or she interacts with the participant.

3. When the claim *Niagree* is satisfied, the Protocol guarantees the participant one-way authentication with *non-injective agreement* with the other participant on some data, provided whenever the protocol has been completed, as the participant believes, with the other participant in the role of the responder:

    - participant receives confirmation that he was actually a participant of this protocol;

- the participant has played the role of responder;
- both participants agree on which data set they used in the exchange.

4. The claim *Niagree* means that if the protocol provides unilateral authentication with the data consistency at all steps of the protocol execution, then we can talk about performing a full consistency.
5. The claim of synchronicity *Nisynch* states a general consistency, but additionally requires that the event of receiving of each message was preceded by the event of sending this message.
6. These definitions allow to divide the process of checking the feasibility of mutual authentication from [5] to sequential checks of the above security properties. This mutual authentication is performed in the case the *Nisynch* claim is satisfied for both participants of the Protocol.

The results of the analysis are shown in Fig 2.

Claim				Status		Comments	Patterns
VUTC	VU	VUTC,VU1	Secret kt	Ok		No attacks within bounds.	
		VUTC,VU2	Secret LOW(KDF(PI(ECP(kb,ECP(kt,P))),TC,VU))	Ok		No attacks within bounds.	
		VUTC,VU3	Alive	Fail	Falsified	At least 3 attacks.	3 attacks
		VUTC,VU4	Weakagree	Fail	Falsified	At least 3 attacks.	3 attacks
		VUTC,VU5	Nisynch	Fail	Falsified	At least 2 attacks.	2 attacks
		VUTC,VU6	Niagree	Ok		No attacks within bounds.	
	TC	VUTC,TC1	Secret kb	Ok		No attacks within bounds.	
		VUTC,TC2	Secret LOW(KDF(PI(ECP(kt,ECP(kb,P))),TC,VU))	Ok		No attacks within bounds.	
		VUTC,TC3	Alive	Fail	Falsified	At least 3 attacks.	3 attacks
		VUTC,TC4	Nisynch	Fail	Falsified	At least 2 attacks.	2 attacks
		VUTC,TC5	Niagree	Ok		No attacks within bounds.	
		VUTC,TC6	Weakagree	Fail	Falsified	At least 3 attacks.	3 attacks

Figure 2: The results of the analysis using *Scyther*.

The results of the analysis of the protocol model, shown in Figure 2, demonstrated a formal violation of the claims *Alive, Weakagree* and *Nisynch*, while the claim *Niagree* was satisfied.

A formal violation of the security claims *Alive* and *Weakagree* is due to the fact that the offender can initiate the start of the protocol by sending a message *GET_CHALLENGE*.

The protocol does not provide the participants (TC and VU) with a confirmation that the participant initiated the start of the protocol is really a legal participant, thus setting up a formal violation of the above claims.

Similarly, the *Nisynch* claim can be formally violated when the offender sends the message *GET_CHALLANGE* at the first step of the protocol. At the same time, further data exchange will be carried out between legal participants (*TC и VU*). In this case, the assumption about legal participant is formally violated.

From the graphic representation of some of the formal security violations found by *Scyther*, it is clear that they occur due to the fact that the on-board device sends a request *GET_CHALLANGE* to the tachograph card. In this case, the directed request does not contain any data used in the protocol further on. The presence of this request is due to the design features of information exchange between the on-board device and the tachograph card.

The violations of the security found with the *Scyther* tool do have a formal character: these violations do not allow the offender to obtain any additional information and do not have any impact on the further execution of the protocol. In this regard, it can be considered that despite the formal violation of some security properties, the investigation of the protocol using the *Scyther* tool also did not reveal the threat of generating a common with a legal participant key or pretending as a legal participant of the protocol.

4. Conclusion

The analysis of the security threat model for the protocol generating the common key with the subscriber authentication intended for use in the tachographs installed on vehicles [3] shows that the studied protocol provides sufficiency of measures of counteraction to the known attacks. The found possible attacks are of a formal nature, not allowing the offender to obtain any additional information to discredit the protocol.

References

[1] European Agreement concerning the Work of Crews of Vehicles engaged in International Road Transport (AETR). economic commission. Inland transport Committee. Note by the Secretariat. Appendix 1B to Annex AETR, which contains requirements for the design, testing, installation and inspection of a digital control device used in road transport. – ECE/TRANS/SC.1/2006/2/Add.1. – 2008.

[2] Order of the Ministry of transport of Russia of February 13, 2013 № 36. El. resource http://www.mintrans.nso.ru/sites/mintrans.nso.ru/wodby_files/files/wiki/2014/12/prikaz_36_13.pdf.

[3] Rosstandart. Information technology. Cryptographic protection of information. Recommendations for standardization. Principles of development and modernization of encryption (cryptographic) means of information security. – 2016. – 36 pp.

[4] Rosstandart. Information technology. Cryptographic protection of information. Recommendations for standardization. Cryptographic authentication mechanisms for use in control devices that ensure the operation of vehicles (draft second edition). – 2017. – 21 pp.

[5] Gorbatov, Victor S.; Zhukov, Igor Y.; Murashov, Oleg N.. Authentication and common key generation cryptographic protocol for vehicle tachographs. IT Security (Russia), [S.l.], v. 24, n. 4, p. 27-34, nov. 2017. ISSN 2074-7136. Available at: <https://bit.mephi.ru/index.php/bit/article/view/274>. Date accessed: 09 aug. 2018. doi:http://dx.doi.org/10.26583/bit.2017.4.03.

[6] van Oorschot P.C., Wiener M.J., Parallel Collision Search with Cryptanalytic Applications// Journal of Cryptology — Vol. 12 — 1999. — 1-28 pp.

[7] Blanchet B., An Efficient Cryptographic Protocol Verifier Based on Prolog Rules, Proceedings of the 14th IEEE Computer Security Foundations Workshop (CSFW), Cape Breton, IEEE Computer Society, 2009, pp. 82–96.

[8] Armando A. et al., The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. Proceedings of Computer Aided Verification'05 (CAV), Vol. 3576 of Lecture Notes in Computer Science, Springer, 2005, pp. 281-285.

[9] Available at: http://www.avispa-project.org/.

[10] Cheremushkin A.V. Kriptograficheskie protokoly: osnovnye svoystva i uyazvimosti [Cryptographic Protocols: Basic Properties and Vulnerability]. Moscow, Akademiya, 2009.

[11] Cremers C. J. F. Scyther - Semantics and Verification of Security Protocols// Ph. D. dissertation. Eindhoven University of Technology, 2006.

[12] Lowe G. A hierarchy of authentication specifications. In Proc. 10th IEEE Computer Security Foundations Workshop (CSFW), pages 31-44. IEEE, 1997.

[13] Blanchet B., An Efficient Cryptographic Protocol Verifier Based on Prolog Rules, Proceedings of the 14th IEEE Computer Security Foundations Workshop (CSFW), Cape Breton, IEEE Computer Society, 2009, pp. 82–96.