# C-ITS Platform

## WG5: Security & Certification

## Final Report

# ANNEX 3: Crypto Agility / Updateability in in Cooperative-Intelligent Transport Systems (C–ITS)

v1.1

Contents

# 1 Scope

This document analyses the topic of crypto agility and updateability in in C-ITS. This report aims to map the current positions of C-ITS Platform WG5 experts on how crypto agility and general updateability can be handled when deploying C-ITS systems.

# 2 Introduction

## 2.1 Definition of the concepts

Crypto agility is the ability of a protocol to adapt to evolving cryptography and security requirements (from RFC 6421). In this report, this definition of crypto agility and general updatability is expanded to the trust model of C-ITS (rather than the single protocol) and also includes the capability to support software updates, modification of cryptographic keys (size of keys), format of certificates, and algorithms during lifetime of the PKI and security protocols. The capability to support software updates is included because it is linked to crypto agility.

## 2.2 Drivers for crypto agility and software updateability

During the work of the EU C-ITS deployment platform's work on security and more specifically on the trust model and the related security structures, WG5 experts raised the question on what happens if there is a need to update the algorithms or the software in the C-ITS station of elements of the E-SCMS (European Security Credential Management system) for various reasons (see section 5). For example, if the cryptographic algorithms are broken, an attacker can masquerade as a CA and issue unlimited certificates. They are also able to forge the signatures on the CRL, which would allow them either to revoke valid CAs or to revoke the CRL signer itself. A broken algorithm would effectively make the system non-secure.

First, one needs to understand when the need for crypto agility might occur. This is the case when the algorithm to create the signatures is broken, (i.e., it is possible to create forged signatures for arbitrary messages) or when the encryption/decryption functions are broken (e.g., it is possible to easily access and read the content of encrypted data). Assuming the chosen algorithm for the signatures at the time of design is state of art, it is possible to select algorithms that with high probability will not be broken simply by more computing power becoming available. However, advances in mathematics or computing technology (e.g., quantum computers) could increase the risk of breaking cryptographic algorithms. These advances are hard to foresee and as such it will be very difficult to predict a timeframe where this might happen (see the section 6.1.2 for details). In addition to the advances in mathematics or computing technology, new weaknesses, new attacks and increase of computer performance can make the security credentials, trust model and security mechanisms vulnerable.

In general, failures in the crypto algorithm should be treated differently from other defects like the failure of a specific C-ITS station or a group of C-ITS stations. The failure of the crypto-algorithm can impact the security of all the C-ITS stations in the overall C-ITS infrastructure but the probability to happen is limited. Instead the failure of a C-ITS station has a limited impact on the C-ITS infrastructure because the C-ITS station can be disconnected or shutdown from an administrative point of view but the probability to happen is higher.

Most attacks and security vulnerabilities in practice are not due to broken cryptographic algorithms, however, but to implementation mistakes that allow the attacker to circumvent the cryptography. These vulnerabilities are much more common than cryptographic breakthroughs, but are no less dangerous for the

security of the overall system. None of the most devastating and advertised attacks of the past years (e.g., HeartBleed, Shellshock, and LogJam) involved any cryptographic breakthroughs, but they were nonetheless detrimental to the security of the affected systems. Updateability of system software to fix faulty implementations is therefore at least as important as being able to replace broken crypto algorithms.

If a migration becomes necessary, how is it to be realized? Two important factors are noted here.

First, the algorithms selected for day-one deployment in C-ITS will be selected on the basis of various needs, including the implementation and deployment costs, as adoption of C-ITS depends on relatively low additional cost to equip vehicles and infrastructure with C-ITS technology, but still providing a sufficient level of security. This means that it would not be reasonable to expect that the hardware installed in the day-one implementation has significantly more computational power than required to support the day-one signature algorithms. Any new algorithm to replace an old one may very well require more computational power, bandwidth, or memory than the broken algorithm: a longer key might be needed, or a new algorithm with fundamentally different mathematics. It is not clear that day-one devices or the G5 channel can support cryptographic algorithms with significantly greater processing or channel requirements than the day-one algorithms.

Second, experience shows that recalls of equipment will never reach 100% and it is extremely difficult to predict timescales for updates, even for equipment connected through wide-area networks that typically require subscriptions, as coverage of those never are 100% and users might have opted out of extension of their subscriptions. The consequence is that there will always be a period where some equipment adheres to the old regime and other to the new. In such a transition period the assurance level of the non-updated equipment could be demoted.

Third, one needs to understand the consequences of a broken algorithm versus the cost of deploying a new algorithm. As C-ITS send the information in clear, the basic consequence of a broken algorithm is that the trust in the C-ITS system cannot be guaranteed any longer. For example, it is possible to fake a C-ITS station. On the other hand, when assessing the consequence of a broken algorithm, it is clear that the effort to produce false ITS stations needs to be weighed against the effort required to take control of the infrastructure and existing ITS stations. In other words, attacks against the cryptographic algorithms must also be balanced against other potential threats to the C-ITS system such as hijacking of the C-ITS stations or update and activation of malicious software.

Fourth, the use of software has increased considerably in recent years in the automotive market[1]. Enabling software update interfaces to components of cars also opens a new attack surface with new risks and threats, which are similar to the one affecting the computer industry (e.g., virus, worms). Appropriate measures must therefore be taken to protect and authenticate software updates.

To summarize, the following threats can be identified, which drive the need for crypto-agility and software updatability:

1. The unforeseen breach of one or more cryptographic algorithms used in the C-ITS trust model.

[1] http://www.wired.com/2011/04/the-growing-role-of-software-in-our-cars/

2. The intentional and planned migration of one or more cryptographic algorithms due to a change in the operational and security requirements of the C-ITS system or application. This may prompt an increase of the size of keys or an adaptation of the certificates formats.
3. The need to update the software operating on a C-ITS station, system or application.
4. The detection of malicious software on a C-ITS station, which requires the removal of the malicious software and the installation of a cleaner version.

# 3 References


| | |
|---|---|
| [1]. | C2C-CC: PKI Memo V 1.7. C2C-CC, "C2C-CC public key infrastructure memo," CAR 2 CAR Communication Consortium, Tech. Rep., February 2011. |
| [2]. | ETSI TS 103 097 Intelligent Transport Systems (ITS); Security; Security header and certificate formats, v1.1.1, April 2013. |
| [3]. | Migration Slide provided by Referat F4 – Kooperative Verkehrs- und Fahrerassistenzsysteme,<br><br>Bundesanstalt für Straßenwesen (BASt) |
| [4]. | IETF RFC 7696. Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms |
| [5]. | 'Public Key infrastructureand crypto agility concept for intelligent transportation systems', M. Ullmann, C. Wieschebrink and D. Kugler, VEHICULAR 2015: the fourt-h international conference on advances in vehicular systems, technologies and application |

# 4 Glossary

| Abbreviation | Synonym | Description |
|---|---|---|
| ASICS | Application specific integrated circuit | Integrated circuit designed for a specific application and therefore not adaptable to new design or changes. |
| Authenticity | Security property | Property that an entity is what it claims to be (ISO 27000). |
| CA | Certificate Authority | The CA is a trusted party, which authenticates entities taking part in an electronic transaction. To authenticate an entity, the CA issues a digital certificate. This certificate is a digital document which establishes the credentials of the entities participating in a transaction. |
| Certificates | Security material | A set of data that uniquely identifies an entity, contains the entity's public key and possibly other information, and is digitally signed by a trusted party, thereby binding the public key to the entity. |
| C-ITS | Intelligent Transport Systems | Intelligent Transport Systems (C-ITS) are systems to support transportation of goods and humans with information and communication technologies in order to efficiently and safely use the transport infrastructure and transport means (cars, trains, planes, ships). |
| C-ITS Application | | A functional definition of a service provided to an end user, which fulfils specific needs of a user (for example, forward collision warning) |
| C-ITS station | | A collection of (functional) equipment that participate in the provision of C-ITS services at a particular location. An C-ITS station may exist in a vehicle, at the roadside, in a central location such as a Traffic Management Centre, or in a mobile device. It has two meanings: an actual physical device and/or a functional set of services.<br><br>In this report a C-ITS station is the equivalent of an ITS-station defined in ETSI documents. |
| C-ITS station system manager | | This is the entity responsible for managing the C-ITS station from an operational and administrative point of view. This role is equivalent to the role System management defined in ISO/DIS 17427-1: "The role "system management" is responsible for all management activities in the system. It |

| | | |
|---|---|---|
| | | supports both System operation and Policy framework". |
| E-SCMS | European-Security Credentials Management System | Security system design for cooperative vehicle-to-vehicle and vehicle to infrastructure applications |
| HSM | Hardware Security Module | The hardware module in the C-ITS station, which implements the cryptographic algorithms defined in the Trust Model. |
| Misbehavior detection | | Automatic detection of misbehaving device or equipment, possibly resulting in automatic revocation. |
| OTA | Over The Air | Through wireless communication, e.g., Wifi, Bluetooth, mobile networks, or radio. |
| PKI | Public Key Infrastructure | A public key infrastructure (PKI) is the combination of software, cryptographic technologies, processes, and services that enable an organization to secure C-ITS communications and business transactions. |
| Policies | | Rules, practices, regulations, laws, official texts governing specific activities, organizations, agreements. |
| Privacy/Data protection | | Set of rules and policies in a jurisdiction, aiming at protecting sensitive personal data belonging to individuals. |
| Revocation | | Revocation is the act of recall or annulment. It is the reversal of an act, the recalling of a grant or privilege, or the making void of some deed previously existing (source: Wikipedia). |
| Security material | | Collection of cryptographic material (keys, certificates, algorithms, credentials, identifiers) that need to be created, embedded, activated, deactivated and eventually discarded at the end of life of a device. |
| V2I | Vehicle to Infrastructure | Vehicle to Infrastructure communications |
| V2V | Vehicle to Vehicle | Vehicle to Vehicle communications |
| V2X | Vehicle to X | Combination of Vehicle to Vehicle communications and Vehicle to Infrastructure communications. |

# 5 Requirements for Crypto agility

The following main requirements and needs for crypto-agility and updatability are identified:

1. Crypto agility should support a secure update of cryptographic algorithms in C-ITS stations due to planned or unplanned reasons.
2. Software Updateability should support a secure update of new software versions or patches in C-ITS stations due to planned or unplanned reasons.
3. Crypto agility should support backward compatibility. In other words, crypto agility should support different sets of cryptographic algorithms at the same time. Note that backward compatibility must only be for a limited transition period, because the system remains insecure as long as broken signatures are accepted.
4. Updateability should support software compatibility. In other words, new software modules and patches should be able to operate on the same hardware of the replaced software.
5. Software reversibility. If a software update is not successful, the C-ITS station should be able to return to the previous working software version.

# 6 Design Options for Crypto-Agility and Software Updateability

The following design options are identified to deal with crypto agility and updateability for C-ITS deployment:

1. Do nothing. In other word, do not design the system to prepare for updates.
2. Already from day one, equip the C-ITS stations with a second cryptographic algorithm, to be used if the first is broken.
3. Design and prepare a secure over-the-air (OTA) update protocol both for the update of the cryptographic algorithms and the software.
4. Design and prepare a secure software update protocol that must be performed by professionals at a workshop.
5. Design and prepare a secure protocol to exchange hardware modules that must be professionally replaced at a workshop.

These options are not mutually exclusive, in the sense that different options could be chosen depending on the type of security breach and on the affected component. For example, generic security patches of the system software could be distributed through remote over-the-air updates, while updates that involve installing new cryptographic keys must be done in a workshop.

Note that the term “protocol” in options 4 and 5 is not limited to a communication protocol but is meant to include various means and procedures to support crypto agility in the different options, possibly involving manual steps by professionals.

Regarding option 1, this is the least preferred but it may happen if the needed resources for standardization, design, development and deployment are not available to implement crypto agility. When this option is chosen, severe security breaches could compromise the security of the overall C-ITS system and require a recall to upgrade all C-ITS components and systems. Given the long lifetime of vehicles and the fast pace at which new software vulnerabilities are found and exploited, it is unrealistic to assume that software installed in vehicles on day one will remain secure until the vehicle reaches the end of its useful life.

Regarding option 2, the possibility of having a second, initially dormant crypto algorithm implemented in the C-ITS station from the beginning may only bring a limited benefit: if the first algorithm is state of the art at

the time of its design, then the second algorithm may not have a significantly longer lifetime until it can be broken, or it might even be broken before the first algorithm. However, this option could make sense if the second algorithm is more secure but has practical disadvantages, such as poor performance or longer signature sizes. Use of the second algorithm would therefore mean that the frequency of status messages must be lowered or less bandwidth remains available for useful content. In such a case, one could use the primary algorithm as long as it is secure, but switch to the second algorithm with reduced functionality in case it is broken.

The main advantage of option 3 is clearly that vehicles and infrastructure can be updated remotely when they have network connectivity, resulting in a faster and more widespread roll-out of the update. Note that vehicles may need regular network access anyway to obtain fresh pseudonyms and updated revocation lists. Whether traffic infrastructure will be equipped with network connectivity will depend on the cost of providing such connectivity versus the cost of manual updates.

A disadvantage of option 3 is that the update interface creates an additional attack surface. A secure protocol must be defined to support the secure download and activation of new software on the C-ITS stations. Note that the secure download of the software requires a trust anchor, which is provided by the trust model. However, this means that a breach of the cryptographic algorithms therefore also impacts the update mechanism. While there are benefits in selecting the same algorithms and suites for different protocols, e.g., because it simplifies the implementation when more than one of the protocols is used in the same device or system (from [4]), it can also be a significant weakness in case that algorithm is broken. As software updates are not subject to the same stringent bandwidth and performance requirements as the small, high-frequency CAM/DENM messages, it may make sense to use stronger cryptography to authenticate software updates..

The main disadvantage of options 4 and 5 is that updates spread much slower, as vehicles need to be serviced by professionals in a workshop to apply the update. Updates will therefore take several months or even years to reach most vehicles, while some vehicles may never receive the update at all unless required by the regulations in place. The advantage of option 4 over option 5 is that the update is easier and less costly to apply as no new hardware needs to be purchased and installed.

Updates to the cryptographic algorithms and key lengths must be introduced in a backwards compatible manner so the new and the old algorithm and/or key length can co-exist during a limited transition period, or so that messages signed with the old algorithm can be demoted to a lower trust level. The standards could foresee open-ended extension fields that can be used in the future for new cryptographic algorithms. This may require a standardisation effort before the first equipment is deployed. It is difficult to assess how feasible this solution is before the work is started, as there are limitations on the size of the message etc. For example, it might be difficult to transmit both the old and the new certificates in the same message.

There can be different designs of the protocols for Options 3, 4, and 5. The following considerations/suggestions are presented:

- The protocols should be modular, using algorithm identifiers to identify new algorithms without requiring the version number to be incremented. This is already the case in TS 103 097 and in IEEE 1609.2, although only one algorithm has been issued an identifier in each case
- The platforms for signature verification should ideally be designed to be reconfigurable.
- The OS and applications on the C-ITS stations should support Over The Air (OTA) firmware upgrade. The upgrade should be authenticated with a crypto algorithm...

- When new crypto algorithms are installed, there must also be a protocol to provide C-ITS stations with new keys for the new algorithms, as the old keys will almost certainly not be compatible with the new algorithms. A protocol should be prepared that will allow new key material to be imported securely in the future, for example by encrypting and authenticating the new keys with symmetric keys that are installed into C-ITS stations by the manufacturer.
- ASICs should not be exclusively relied on for private key operations, although they are deemed more secure than software implementations. HSMs should preferably be software-based and support firmware upgrade where the firmware is signed with a signature algorithm. Alternatively the C-ITS station system manager will have the burden to re-call and exchange the HSMs.
- Customized ECC ASICs for cryptographic acceleration will not be able to accelerate non-ECC cryptography. Thus the system should be designed so that software implementations of verification can be used and/or added to hardware acceleration. Software implementation can be used when performance requirements are not critical. For example, software implementation of cryptographic algorithms may not be recommended for signing CAM/DENM messages but it could be feasible to sign the download and activation of new software. An interesting observation arising from this is that even on the ECC side, curves that are easy to make constant-time in software should be favoured over curves that are hard to make constant-time in software but easy to make constant-time in hardware.
- It is commonly expected that quantum computers will break all elliptic curve algorithms one day. The NSA and the EU have advised organizations to have a post-quantum cryptographic strategy in place by 2020.
- There is a higher risk that CA certificates will be hacked than short-term certificates, because the likelihood and impact is much higher. Therefore the robustness requirements to sign CA certificates is much higher than for short-term certificates. Correspondingly, it is more important to be able to update the verification algorithms than the signature algorithms of ITS-Stations.

In addition, security recommendations for crypto agility can only be valid for a limited time and have to be reassessed on a regular basis.

For example, with a scheme like depicted in **Figure** 1, based on a limited lifetime of the Root certificates and all the elements on lower levels depending on the Root CA, "old security principles" can be phased out and replaced by new ones during an appropriate overlapping time.

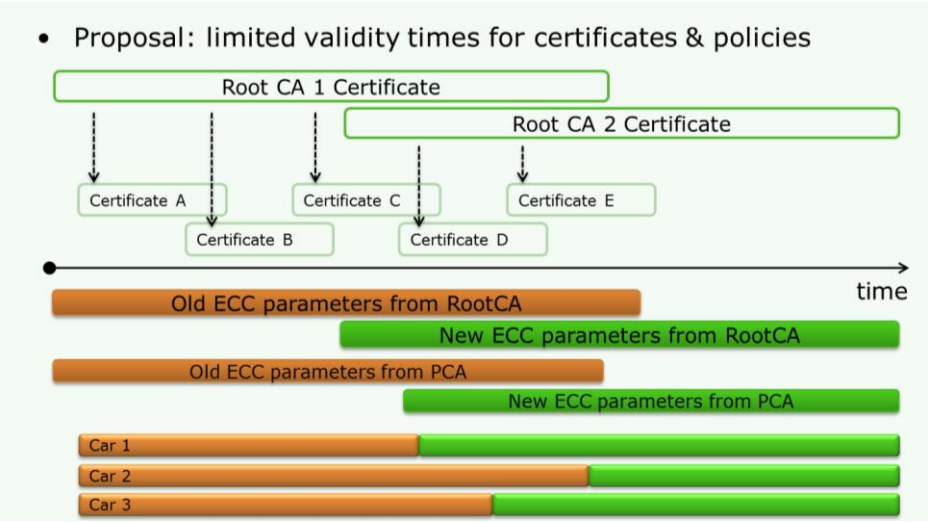


**Figure 1 Crypto-agility approach based on limited validity times of the certificates**

### 6.1.1 Issues and considerations for elliptic curve cryptography

The family of cryptographic approaches known collectively as elliptic curve cryptography (ECC) forms the basis for all currently recommended C-ITS-specific communications security. In ECC, operations are carried out on *points* on a particular *curve*. The details of the operations and the curves are not important to this discussion other than to note that different operations and curves have different performance and security properties. Specific curve types are known as Weierstrass, Montgomery, and Edwards; specific curves that are widely used in cryptography are known as the NIST and Brainpool curves (which are both Weierstrass curves), curve25519 (a Montgomery curve, used for key exchange) and ed25519 (an equivalent Edwards curve, used for signatures).

The NIST and Brainpool curves are more than 15 years old, and more recent research has highlighted the Edwards and Montgomery curve families as providing faster operations. In particular the Montgomery curve called Curve25519 has gathered quite a lot of support, being efficient, relatively straightforward to protect against side channel attacks in software, straightforward to implement without exploitable flaws, and "demonstrably random" in the same sort of the way that the Brainpool curves are (from ETSI SAGE). This curve is included in some publicly availably libraries, is used in TLS, and has been recommended for use in future IETF standards, but it is still not widely adopted in other standards bodies such as ISO. However, for certain hardware multipliers, a commonly-used technique for blinding against power consumption and timing attacks results in private key operations on these curves running slower than the corresponding operations on the Brainpool curves.

ENISA has also made proposals for 'future proof' solutions, e.g. EC-Schnorr or ECKCDSA (ref: Algorithms, key size and parameters report – 2014).

### 6.1.2 Cryptographic algorithms resistant to quantum computing

As described before, it is commonly expected that quantum computers will break all elliptic curve algorithms one day. The NSA and the EU have advised organizations to have a post-quantum cryptographic strategy in place by 2020. This leads to the observation that any ECC algorithm beyond 256-bits does not provide protection against quantum computers, only against regular increase of computation power as long as quantum computers are not available. On the other side, there will be some time to prepare for the transition: quantum computers are not yet developed and even if developed not easily available to hackers.

For quantum safe cryptography solutions, there are a lack of clearly established standards esp. for digital signature and we should clearly distinguish between time-line for future solutions (e.g. current ECC technologies used in our C-ITS standards, step2 with more future-proof solutions, step3 post-quantum solutions). There is of course an issue of technology switching when the quantum computers will become available, but with the state of the art in crypto today there are too much risks of selecting quantum-safe technologies which are not so much publicly available in SW libraries or chips, not widely adopted standards, bringing risks that they are not safe, and not cost-affordable.

It is noted that a number of projects develop quantum-resistant cryptography, in particular the two Horizon 2020 projects pqcrypto (http://pqcrypto.eu.org) and SAFEcrypto (http://www.safecrypto.eu/). Additionally, the ETSI Quantum-Safe Crypto (QSC) working group is developing a framework for assessing and recommending quantum-safe algorithms.

It is further noted that ISO/IEC JTC 1/SC 27/WG 2, in charge of Cryptography & security mechanisms, has started the Study Period on "Quantum computing resistant cryptography" from May 2015.

# 7 Recommendations

The following recommendations are identified by the C-ITS Platform WG5 experts:

(1) The responsible policy body for the definition of the security policy shall be in charge of defining updates of the security policy due to expiry/deprecation of crypto algorithms, and a migration plan that shall be observed by the C-ITS Station system manager.

(2) It is recommended to investigate and potentially amend the existing protocol defined in ETSI to allow, in a backwards compatible manner, a second signature, which uses a new algorithm and which can co-exist with the old signature during a certain period. The design of the protocol should take in consideration the analysis provided in section 6.

(3) C-ITS Station system managers shall estimate the risks related to certificate policy (e.g., new cryptographic algorithms) and software updates and have an appropriate risk treatment plan.

(4) Security recommendations on cryptographic algorithms for C-ITS can only be valid for a limited time and have to be reassessed on a regular basis.

(5) It is important to define a framework to reach quickly a consensus on the choice of new algorithms in case of a security breach. In other words, an organization and process should be put in place to reach quick solutions to security breaches in the cryptographic algorithms.