The Crypto-Agility Properties

Hassane Aissaoui Mehrez
Network and Computer Science Department
INSTITUT MINES TELECOM-Télécom ParisTech
75 013 Paris, France
hassane.aissaoui@telecom-paristech.fr

Othmane EL OMRI
IRT SystemX Department
SystemX Institute of Research & Technology
Paris Saclay, 91120 Palaiseau France

ABSTRACT

Due to the continuous population growth and the size of urban areas, transportation systems are facing significant challenges: traffic congestion, innovative real time travel information services, etc. Intelligent Transportation Systems (ITS) makes a positive contribution in addressing those challenges, by working towards alleviating congestion, maintaining a high level of operational efficiency in the transportation system, and predicting and preventing traffic incidents, as to its social, economic and environmental impact.
Along these lines, ITS aim at providing sophisticated management models and tools across all transport modes. Therefore designing models that resist future attacks is an important area of research and standardization, in particular in our project Secure Cooperative Autonomous systems in relation to the domain of Cooperative Intelligent Transport Systems. So it is far from easy due to the high complexity of these systems. To preserve the security properties of today's communications against future attacks, protocols must begin to explore the new challenges to design and implement crypto-agility, which has the ability to switch easily from an obsolete cryptographic algorithm to another.
To achieve these challenges, we introduce properties for the crypto-agility in order to allow designers to develop generic models, while providing more agility and significantly reducing the development cycle.

Keywords: Post-Quantum Cryptography, Public-Key Encryption, C-ITS, HSM, crypto-agility.

1. INTRODUCTION

Today's vehicles are already connected devices. Tomorrow, they will also communicate directly with each other and with the road infrastructure. This communication will allow road users and traffic managers to share information and use it to coordinate their actions.
This coordination - between vehicles and between vehicles and transport infrastructure - is expected to significantly improve road safety, driving comfort, helping the driver to take the right decisions, etc.
However, to increase the safety of future cooperative vehicles, many security requirements should be guaranteed at different levels. This is far from easy due to the high complexity in the domain of C-ITS.
In this paper, we will discuss our motivations (§2). In (§3) we relate some works and the best practices to achieve crypto-agility. In (§4) we define crypto-agility and we detail our contribution. We finish this paper with a conclusion and perspectives.

2. THREATS OVERVIEW AND MOTIVATIONS

Motivations

The main motivation for our project is to maintain the security system and on the long-term. Having a fixed security system is a bad approach since what is secure today can be insecure tomorrow, therefore the need to quickly and efficiently update the security system will always be present. There are several motivations to update security systems and encryption algorithms in particular:

- The discovery of new attacks and cryptanalysis techniques,
- The constant evolution of computing performance according to Moore's law which lowers the security of the existing cryptographic algorithms. For example, attacks like brute force will take less time to find the secret key,
- The development of quantum computers which is threatening the security of several cryptosystems. This topic is discussed in more detail in the following sections,
- If the lifecycle of cryptographic systems (microchips, secure elements, embedded HSMs) exceed ten years, then they will likely be decrypted, compromised or cloned in the next decade.

In general, without crypto-agility, updating or changing the security system can be a very slow and inefficient process. In fact, for an implementer, this would mean changing the code, rebuilding it, testing it, deploying patches to all the users, etc. Not only does this process lack efficiency but it can also be a

big window of attack if one of these steps is not followed properly.

Threats Overview

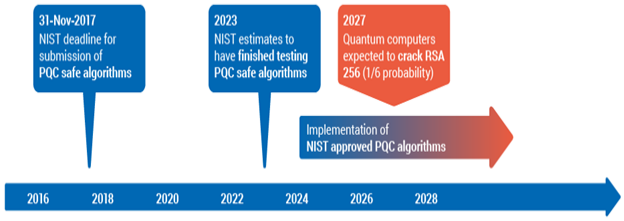
Recent advances on the development of quantum computers are more than ever threatening the majority of cryptographic algorithms used nowadays. Therefore, they constitute a major motivation for having crypto-agile systems that can change efficiently their crypto algorithms in case they are broken due to the computing capacity of these computers.
The quantum computers are not "better" than classic computers in an absolute way, they just specialize in different tasks because they are built following a completely different architecture than classic computers. In fact, they use quantum phenomena such as superposition of states and entanglement of particles to perform computations. Therefore, there can perform some tasks in a much more efficient way than classic computers.
In fact, in quantum computers the information is represented by "quantum bits" (qubits), which is the quantum equivalent of the binary bits used in conventional computing. A bit can take two logical values (0 or 1). The qubit however, is a probabilistic unit of information that can be in a superposition of both states (0 and 1).
Quantum computers are very good at factorizing numbers into prime factors using Shor's algorithm, which is an algorithm specially designed for quantum computers. Shor's algorithm can factorize a number in a polynomial time, while the best factorization algorithms for classic computers run in exponential time. Therefore, quantum computers can easily break any cryptosystem that relies on the difficulty of factorization such as RSA.
Moreover, a modification of Shor's algorithm can be used to solve the discrete logarithm problem, which is a difficult problem for classic computers, and a lot of asymmetric cryptosystems rely on the difficulty of this problem such as ElGamal, ECC (Elliptic Curve Cryptography), Diffie-Hellman key exchange protocol, DSA, etc. Hence, the security of these widely used cryptosystems will be broken when quantum computers are available on a large scale.

Table 1 Key strength for conventional computing [1]

Algorithm	Effective Key Strength / Security Level		
	Key Length	*Conventional Computing*	*Quantum Computing*
RSA-1024	1024	80	0
RSA-2048	2048	112	0
ECC-256	256	128	0
ECC-384	384	256	0
AES-128	128	128	64
AES-256	256	256	128

Another thing that quantum computers are good at is searching over a set of data. Given a black-box function f, Grover's Algorithm is an effective way to invert the function f, which means search for x such as f(x) = y when given y. This algorithm halves the time needed to perform this operation in comparison with a classic computer, this means that symmetric algorithms such as AES or hash functions are made more vulnerable. More specifically, their security is divided by two. However, not all is lost and the general recommendation is to double the size of keys, for example AES-512 should be used to have the same level of security as AES-256.
"Table 1," shows the effective security level of different crypto algorithms with conventional computers and quantum computers. A security level of n bits means that breaking the algorithm will require performing on average 2^n operations.
If the cryptographic systems are based on any of the above, and their lifecycles exceed ten years, then microchips, secure elements, embedded HSMs, generated with the affected algorithms, will likely be decrypted, compromised or cloned within the next decade "Figure 1,".

Figure 1 Quantum computer expected timeline [2]



Due to this concern, many researchers have begun to investigate the crypto-agility design and implementation. The goal of this research is to develop cryptographic algorithms that would be secure against both classical and quantum computers, and could serve as counter measures for the eventuality that large-scale quantum computers become a reality. NIST (National Institute of Standards and Technology) has decided that it is wise to begin developing standards for post-quantum cryptography, because it appears that a transition to post-quantum cryptography will not be simple. Therefore, it is desirable to plan for this transition early.
NIST (National Institute of Standards and Technology) has decided that it is wise to begin developing standards for post-quantum cryptography, because it appears that a transition to post-quantum cryptography will not be simple. Therefore, it is desirable to plan for this transition early.
Currently, there are several post-quantum cryptosystems that have been proposed, including lattice-based cryptosystems, code-based cryptosystems, multivariate cryptosystems, hash-based signatures, and others.
However, for most of these proposals, further research is needed in order to gain more confidence in their security and to improve their performance.

3. RELATED WORKS AND BEST PRACTICES

There are a few documentations that address the topic of crypto-agility such as the RFC 7696 [3][4] and the C-ITS Platform Final Report [5]. Moreover, a few existing security protocols have some crypto-agility built in. For instance, the popular web security protocol TLS/SSL allows for cipher suites to be changed, and X.509 digital certificates can support new cryptographic algorithms.
The RFCs (7696/6421) described the processes and perspectives of recent guidelines from the Internet Engineering Task Force (IETF) and described the requirements for a crypto-agility solution.
The goal of RFC 7696 is simply to state the obvious but not easy to achieve: to ensure that security protocols can migrate

from one algorithm or suite of algorithms to a newer, stronger one when needed.
Addressing this, RFC 7696 recommends viewing the problem from two perspectives: that of the protocol designer and that of the protocol implementer.
Protocol designer: The protocol designer must identify which algorithms are old/new, and which are mandatory to implement to achieve interoperability.
Protocol implementer: The protocol implementer needs to be able to add a new algorithm, which requires modularity in the cryptographic system, and they need a way to detect if the new algorithm has been successfully added, which is something currently missing from many protocols.
To ensure cryptographic agility and to allow new cryptography to be substituted at some point in the future, then there are many choices that system designers can use.
In general, the common principles for designing a crypto-agile system are:

- Plan for crypto-agility in the earlier steps of design,
- Assume the crypto algorithms will be broken in the system's lifetime,
- Implement the crypto algorithms in a crypto-agile manner,
- Allow efficient change of algorithms in a fully deployed system.
- The C-ITS Platform [5] discusses the issue of crypto-agility in the C-ITS context. The report identifies the motivations of crypto-agility, the requirements and the threats on the C-ITS system, and considers a few design options to deal with crypto-agility. The main options discussed are as follows:
  - ✓ As of day one, equip the C-ITS stations with a second cryptographic algorithm, to be used if the first is broken,
  - ✓ Design and prepare a secure over-the-air (OTA) update protocol both for the update of the cryptographic algorithms and the software,
  - ✓ Design and prepare a secure software update protocol that must be performed by professionals at a workshop,
  - ✓ Design and prepare a secure protocol to exchange hardware modules that must be professionally replaced at a workshop.

1) Option 1 is the least preferred,

2) Option 2 is not sufficient because the two algorithms could be broken given enough time. One algorithm may have longer lifetime than the other but there is no guarantee that it will stay unbroken during the lifetime of the C-ITS station. This solution is not flexible enough.

3) Option 3 has the advantage that the updates could be rolled-out in a faster and more widespread way, if the required infrastructure is available, and the C-ITSs are connected to the network. However, it creates an additional attack surface. If the cryptographic algorithms are broken, those cannot be used to secure the update. Either the C-ITS will not be updated, or it will be updated using an update that could be malicious and create even more issues.

4) Option 4 & 5 are safer and more trustworthy ways to update vehicles, but they can be very slow as vehicles need to be serviced by professionals in a workshop to apply the update. Option 5 is more costly than option 4 because it needs new hardware for every update.
The final choice is not necessarily only one of these options, it can be a combination of different options. For example, OTA updates can be used when the algorithms used in the secure update protocol are still secure. When these algorithms are broken, then it becomes necessary to go to an official workshop in order to update the vehicle.
The protocols should be modular, using algorithm identifiers to identify new algorithms without requiring the version number to be incremented [6] [7].
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.
Other recommendations deal with hardware crypto-agility: ASICs (application-specific integrated circuit) 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.
The report also pointed out the recommendation by the NSA and the EU to have a post-quantum cryptographic strategy in place by 2020.
The paper [8] discusses the issue of crypto-agility in a PKI system. The goal is being able to change the cryptographic algorithms or parameters such as the elliptic curve domain parameter, the hash algorithm or the signature algorithm.
As mentioned before, the design of a crypto-agile system should be considered at different levels and should be modular. Along these lines, we recommend viewing the problem from another perspective. In fact, we propose some properties of the crypto-agility and we define it. This contribution is outlined in § (4.2).
Before defining these properties, we define what crypto-agility is?

4. CRYPTO-AGILITY PROPERTIES

Definition of Crypto-Agility

A large number of protocols use cryptographic algorithms to guarantee the properties of security (confidentiality, integrity, authentication, digital signature, etc.). Unfortunately, these cryptographic algorithms get older and weaker over time. Progressively as new cryptanalysis techniques are developed or new technologies have improved their computational capacity, the time required to break a particular cryptographic algorithm will be reduced, which will make the attack of the algorithm easier for more attackers.
Therefore, protocol designers need to consider these technological advances that will eventually make any algorithm obsolete. For this reason, protocols need mechanisms to migrate from one cryptosystem to a more secure one.
This adaptive capacity is called cryptographic agility or crypto-agility. In other words, crypto-agility is the ability of a system to migrate easily from one cryptographic algorithm to another,

in a way that is flexible, scalable, and dynamic.
The NIST [9] highlighted the key principles related to cryptographic services, implementations, and agility. The first is that agility is essential for protecting the systems against future threats and supporting backward compatibility. She outlined three facets of cryptographic agility:

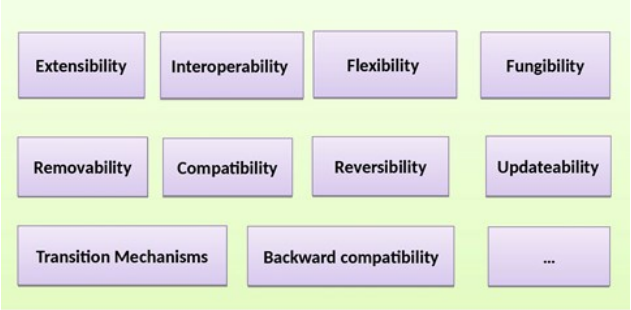
- The ability for machines to select their security algorithms in real time and based on their combined security functions;
- The ability to add new cryptographic features or algorithms to existing hardware or software, resulting in new, stronger security features;
- The ability to easily retire cryptographic systems that have become either vulnerable or obsolete.

It emphasized that the need of agility is also essential for implementations, not just algorithms.
This definition of Crypto-Agility and general update is extended to the C-ITS trust model and also includes the ability to support software updates, modification of cryptographic keys, and security protocols. Therefore, Crypto-Agility needs some important properties for maintaining the security of a system in the long run. In the following paragraph, we will highlight these properties.

Crypto-Agility Properties

To satisfy each security property (confidentiality, integrity, authentication, etc.), we use specific cryptographic algorithms.
In the same way, we defined several properties, which we associated to Crypto-Agility, so we could evaluate each property, and easily determine how crypto-agile a system is.
Indeed, most of the time, adding a new component causes problems and increases confusion. This means that implementations must be modular components that allow protocols, software and systems to easily insert new algorithms or algorithm suites. The main proprieties for Crypto-Agility are identified in "Figure 2".

Figure 2 : Crypto-Agility properties



In the non-exhaustive list below we present some of the most important crypto-agility properties:

Extensibility: it is simply the ability to add new algorithms or new parameters to the system as efficiently as possible. Extensibility is very important especially in case a major flaw is discovered that affects all the algorithms that are implemented in the system at this time. Protocols should be extensible, although extensibility is much harder in practice than it sounds.

Removability: the ability to gracefully retire cryptographic systems that have become either vulnerable or obsolete. This can be tricky since it affects interoperability. Algorithm deprecation has proven to be a slow and inefficient process (MD5, RC4).

Fungibility: the ability for security components to be swapped easily. Fungibility is important for example in the case of algorithm deprecation. Cipher suites are fungible In TLS, because they have an identifier.

Interoperability: Crypto-Agility solutions must be interoperable between independent implementations based purely on the information provided in the specification. Ideally, it is to be able to select the same algorithms and suites for different protocols. Using the same algorithms can simplify implementation when more than one of the protocols is used in the same device or system. Unfortunately, to increase the likelihood of interoperability among diverse devices, sometimes more than one algorithm is needed.
However, interoperability can also complicate agility. It makes it harder, for example, to remove outdated protocols or algorithms such as RC4 (Rivest Cipher 4), which should be removed. Some systems cannot be updated because of hardware restrictions. As a result, interoperability sometimes means supporting algorithms that are not very secure.

Updateability: in many cases, the attacks and vulnerabilities discovered exploit a flaw in the implementation and not in the actual algorithm. Therefore, the updateability is a desirable property in order to support a secure update or patch of cryptographic algorithms in the system. When this situation arises, it is recommended to stop using the flawed algorithm until the update is effective.
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.

Flexibility: In hardware, the cryptography is built in, but with FPGAs, there is slightly more flexibility. Operating systems have even more flexibility, applications can be very agile, and scripting languages such as JavaScript offer tremendous agility.

Compatibility: If we replace software on a system, the new software modules and patches should be able to operate on the same hardware.

Reversibility: If any software update is not successful, the system should be able to return to the previous working software version.

Transition Mechanisms: The negotiation of a cryptographic algorithm should be protected against integrity. Without protecting the integrity of the algorithm, then the protocol will be subject to further downgrade attack opportunities. The transition mechanism is needed to determine whether the new algorithm has been deployed.
Crypto-agility often requires transition periods to change the algorithms or the implementation. A system is considered more agile as the transition period gets shorter. In such a transition period the assurance level of the non-updated system could be

downgraded. That transition must only be for a limited period, because the system remains insecure as long as broken signatures are accepted.

Backward Compatibility: Crypto-agility solutions must be backward compatible with existing implementations. That is, an implementation must support all legacy cryptographic algorithms and must be backward compatible with existing systems.
Backward Compatibility is needed to ease the transition to legacy cryptographic algorithms, it is only appropriate when the new cryptographic algorithms are compromised. Therefore, the cryptographic system 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.

5. CONCLUSIONS AND PERSPECTIVES

In this paper, we discussed how crypto-agility can be essential for protecting against future threats, but can also be extremely challenging. It must be a design technique for allowing protocols, software and systems to replace the cryptographic subcomponents over time. This could be achieved in various ways, if all sub-components of the security system are agile.
To achieve these challenges, we proposed to consider the problem from another perspective. Indeed, our contribution consists in defining crypto-agility properties in order to allow the protocol designers to develop generic models as much as possible to make it easy to change cryptographic algorithms and therefore address many security / safety requirements while providing more agility and significantly reduce the development and deployment cycle.
The perspectives are to analyze these properties in order to establish the interdependence relations between each property. Then, we will define the algorithms or the models that will satisfy each property and we will detail the possible techniques to obtain a crypto-agile system.

6. ACKNOWLEDGMENT

This research work has been carried out within the framework of the Secure Cooperative Autonomous Systems project (SCA), in partnership with Télécom-Paris-Tech and the SystemX Institute of Technology Research.

7. REFERENCES

[1] C.W. Churchman, **The Design of Inquiring Systems**, New York: Basic Books Inc. Pub., 1971.
[2] J. Ivari, "A Paradigmatic Analysis of Contemporary Schools of IS Development", **European Journal of Information Systems**, Vol. 1, No. 4, 1991, pp. 249-272.

[1] NIST SP 800-57, **Recommendation for Key Management,** https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf
[2]https://hsm.utimaco.com/solutions/applications/post-quantum-crypto-agility/
[3] RFC7696 : **Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms**, https://tools.ietf.org/html/rfc7696
[4] RFC6421 : **Crypto-Agility Requirements for Remote Authentication Dial-In User Service (RADIUS)**, https://tools.ietf.org/html/rfc6421
[5] C-ITS Platform WG5 : **Security & Certification Final Report**. **ANNEX 3: Crypto Agility / Updateability in Cooperative-Intelligent Transport Systems (C–ITS)**
[6] ETSI TS 103 097 V2.0.9 : **Intelligent Transport Systems Security : Security header and certificate formats.**
[7] IEEE Standard 1609.2 for Wireless Access in Vehicular Environments: **Security Services for Applications and Management Messages**.
[8] Public Key infrastructure and crypto agility concept for intelligent transportation systems. M. Ullmann, C. Wieschebrink and D. Kugler. VEHICULAR 2015
[9] Cryptographic Agility and Interoperability: Proceedings of a Workshop (2017).