FASTR

Future of Automotive Security Technology Research

Automotive Industry Guidelines for Secure Over-the-Air Updates

FASTR Connectivity and Cloud Work Group

Version 2 — April 3, 2018

TABLE OF CONTENTS

1 INTRODUCTION

What is FASTR? FASTR (Future of Automotive Security Technology Research) is a non-profit consortium focused on accelerating automotive security innovation to enable trust in the autonomous vehicle of the future by catalyzing the creation and deployment of key technologies. FASTR views the automotive security landscape holistically, including everything from the physical supply chain, to consumer electronics used to unlock your car door, to the technical stack responsible for perception and motion planning, and beyond.

During the first half of 2017, the FASTR Connectivity and Cloud Work Group developed guidelines for analyzing secure over-the-air (SOTA) software update systems for the automotive industry. This document, now updated, is intended to provide those evaluating potential solutions with a comprehensive and objective standard by which to analyze SOTA software update systems. Users of this document are encouraged to consider the information provided and to determine how to weight certain guidelines based on their specific implementation or use case.

This document is organized as follows. Section 2 gives definitions for the terms used in the rest of the document. Section 3 discusses the threat model. Section 4 discusses the cryptography used. Section 5 gives guidelines for a key management plan. Finally, Section 6 has a checklist of all the guidelines to aid in evaluating systems.

2 DEFINITIONS AND ACRONYMS

Definitions for terms and acronyms used in this document can be found in Figure 1 below.

SOTA Software Update Definitions and Acronyms	
Term	**Description**
ACL	Access Control List
Adversary	Any entity attacking the software update system in any manner
AES	Advanced Encryption Standard
Backend Services	Services provided by system servers or in the cloud
Denial of Service (DOS) Attack	Attacks in which normal access to the SOTA system is unavailable, typically due to a large amount of malicious activity directed at the system
Device	Any entity being updated, including but not limited to phones, laptops, IoT devices, or automobiles and their components
DRBG	Deterministic Random Bit Generator
ECDH	Elliptic Curve Diffie Hellman

SOTA Software Update Definitions and Acronyms

Term	Description
ECDSA	Elliptic Curve Digital Signature Standard
Escalation of Privileges Attack	Attacks in which an entity obtains additional rights within the system and uses them to perform unauthorized activities within the SOTA system
FASTR	Future of Automotive Security Technology Research
FIPS	Federal Information Processing Standard
Gateway	Device managing the network connecting between the car and the backend services
HSM	Hardware Security Module
IETF	Internet Engineering Task Force
Information Leakage	Unauthorized access to system secrets, confidential data, software and/or IP
KMP	Key Management Plan
LNCS	Lecture Notes in Computer Science
MQV	Menezes-Qu-Vanstone authenticated Diffie-Hellman protocol
NIST	National Institute of Standards and Technology
OEM	Original Equipment Manufacturer
OTA	Over the Air
OTP	One-Time Programmable
PAKE	Password Authenticated Key Exchange
PIN	Personal Identification Number
Repudiation Attacks	Installation of software from unauthorized entities
RFC	Request for Comments
RNG	Random Number Generator
Root Certificate Authority	Organization managing the root signing keys for the system
SHA	Secure Hash Algorithm
Software	Anything being updated, including but not limited to data, configuration fields, an application, firmware, or operating system

SOTA Software Update Definitions and Acronyms	
Term	**Description**
SOTA	Software Over-The-Air
SOTA System	The entire SOTA software update system, including services, client devices, etc.
Spoofing	Installation of software to an unauthorized device or from an unauthorized server
SSL	Secure Sockets Layer
Tampering	Unauthorized modification of device software
TLS	Transport Layer Security
TRNG	True Random Number Generator

Figure 1: SOTA Software Update Definitions and Acronyms

3 THREAT MODEL

SOTA software update systems should be resistant to any attack that does not physically modify the vehicle. Attacks requiring physical modification of any part of the vehicle are considered out of scope. For example, an attack which succeeds in loading unauthorized software through a USB port in the vehicle is considered in scope. If malicious code is running on the vehicle, it should not be able to affect the operation of the SOTA software update system. On the other hand, an attack in which a device in the vehicle is physically modified to allow direct writes to flash memory is considered out of scope. Side-channel attacks such as differential power analysis (DPA) [4] used to extract keys, software, or other sensitive data are also considered out of scope.

[GDL 1] Compliant SOTA software update systems should be resistant to any attacks that do not physically modify the vehicle.

[GDL 2] Side-channel attacks and attacks requiring physical modification of any component of the overall system are considered out of scope for compliant SOTA software update systems.

[GDL 3] Attacks against which compliant SOTA software update systems should be resistant include, but are not limited to:

1. Spoofing attacks
2. Tampering attacks
3. Repudiation attacks
4. Information-leakage attacks
5. Denial-of-service attacks
6. Escalation-of-privileges attacks

The following sections give descriptions of the attacks listed above, as well as some high-level guidelines to defend against these attacks. More detailed recommendations to guard against these threats are given later in the document.

3.1 SPOOFING ATTACKS

Spoofing is an attempt by an adversary to emulate an authorized component of the system. In a SOTA software update framework, this could apply to the device being updated, the server providing the update, any authentication or authorization service such as a Certificate Authority, or any intermediary in any of these processes. For example, an app can act as an intermediary that downloads the update and then sends it on to the device. Spoofing is usually network based, as when a component pretends to a network an identity it doesn't possess. It can also be cryptography based, as when a component convinces another component that it possesses the proper keys for cryptographic signatures.

3.2 TAMPERING ATTACKS

Tampering attacks are an attempt to modify software so that it does something other than what was intended. This can happen at build time, deployment time, in storage, or while running. Often this is achieved by convincing a device to download and use software that has been modified from what the manufacturer intended.

3.3 REPUDIATION ATTACKS

A repudiation attack allows a component to plausibly deny that it executed (or did not execute) some action. For example, a device could claim that it already installed a later revision of software so that the update server does not send it valid updates.

3.4 INFORMATION LEAKAGE

Information leakage encompasses both accidental leakage of information that could be useful to attackers and the intentional bypassing of confidentiality controls. In the scope of SOTA software update frameworks, this will usually mean exposing cryptographic keys or other authentication information or making the software easier to reverse engineer. Sensitive information includes but is not limited to:

1. Secret keys
2. Credentials
3. OEM firmware and software
4. OTA software
5. Vehicle data

3.5 DENIAL-OF-SERVICE ATTACKS

A denial of service is any attack which degrades the availability of a component or of the system overall. The DoS could be against a device, a vehicle the device is attached to, an update server, or a supporting or intermediary system. A DoS attack resulting in loss of network connectivity for any of the components in system is out of scope for the SOTA software update framework. However, it is important that the system degrades gracefully in the presence of such an attack.

3.6 ESCALATION-OF-PRIVILEGES ATTACKS

Escalation of privilege attacks allow an attacker with some level of authorization to perform actions that would normally require a higher level of privilege. This type of attack usually involves compromising a piece of software. In the context of a SOTA software update system, this means the compromise of an in-vehicle OTA agent or a component in the cloud, where the compromised software is used as a mean to access to expand privileges. For example, suppose an agent running as root is compromised. Then it may be able to access R/W areas which the original agent was not authorized to access, such as private user data. The damage from a compromised agent can be limited if it is constrained by an access control list (ACL) to only access a defined set of resources.

3.7 THREAT MITIGATION GUIDELINES

The following are high-level guidelines to help defend against the attacks listed above. More detailed guidelines on implementing these high-level guidelines are given later in the document.

[GDL 4] Compliant SOTA software updates should include a signed certificate containing the public key of the entity providing the update.

[GDL 5] Compliant SOTA software updates should be encrypted.

[GDL 6] Compliant SOTA software updates should be digitally signed, after encryption, with the private key of the entity requesting the update.

[GDL 7] Compliant SOTA software update systems should never install software for which the digital signature is invalid.

[GDL 8] Compliant SOTA software update systems should never install software by unauthorized entities as indicated by the certificate permissions fields (e.g., X.509 Certificate Extensions), even if the digital signature is valid.

[GDL 9] Software updates in compliant SOTA software update systems should include version information to prevent rollback to genuine but obsolete software versions.

Identifying components based solely on network attributes such as IP address, host/domain name, or cellular identifiers is insufficient.

[GDL 10] Compliant SOTA software update systems should secure all network transactions with TLS public key authentication, and the public keys should be signed by a trusted Certificate Authority.

[GDL 11] Use of SSL for network security is discouraged in compliant SOTA software update system.

[GDL 12] Clients in compliant SOTA software update systems should perform host-name verification to ensure they are connecting the correct server. Specifically, they should verify that the host name in the TLS certificate's subject AlternativeName's DNSName field matches the host name they are trying to connect to.

Manufacturers should consider what information might be needed in forensic or crash investigations. All important events should be logged in such a way that they cannot be altered later.

[GDL 13] Designers of SOTA software update systems should seek the recommendations of domain experts to determine what information might be needed in forensic or crash investigations.

[GDL 14] Compliant SOTA software update systems should log all important events, in such a way the log entries cannot be altered later.

The above guideline can be accomplished by writing them to one-time programmable (OTP) memory, or digitally signing the log entries.

It is very difficult to stop a high-capability attacker from acquiring and reverse engineering software, but a manufacturer can raise the bar by distributing software to authorized devices only.

[GDL 15] Compliant SOTA software update systems should deliver software updates to authorized devices only.

Understanding which components can affect others and in what way can help defend against escalation of privileges attacks. Threat-modelling methodologies such as those described in [18] can be extremely helpful.

[GDL 16] Designers of SOTA software update systems should perform threat modeling to determine which components of a system can and should be able to affect other components.

It is difficult to prevent DoS attacks when a system is dependent on an underlying network over which the manufacturer has no control. However, manufacturers should consider the possibility of a DoS and handle it gracefully.

[GDL 17] Compliant SOTA software update systems should be designed to fail gracefully in the presence of a DoS attack.

The following guidelines are to help defend against malicious code being installed, or which may have been previously installed. Whether a component of the SOTA software update system can support these guidelines will, of course, depend on its capabilities.

[GDL 18] Wherever feasible, target devices in compliant SOTA software update systems should be initialized using a secure boot mechanism which can only be modified by authorized entities.

[GDL 19] Wherever feasible, clients in compliant SOTA software update systems should utilize anti-malware protection such as whitelists and in-memory protection.

[GDL 20] If the SOTA software update system uses a shared resource such as a processor, it should be the only process allowed to access those shared resources during the entire software update process.

[GDL 21] Compliant SOTA software update systems should clear all shared resources of sensitive data and keys which were temporarily stored during the software update.

For example, if a hardware accelerator is used to decrypt a software update, the key schedule should be flushed from the accelerator by performing a dummy decryption with random or all zeros data and key.

4 CRYPTOGRAPHIC ALGORITHMS

This section contains recommended cryptographic algorithms for SOTA software update systems. In general, to achieve n bits of cryptographic security, the cryptographic algorithms should have the following key and data sizes.

1. Encryption algorithm: n-bit keys
2. Hash algorithm: 2n-bit message digest
3. Elliptic curve-based digital signature algorithm: 2n-bit keys
4. Elliptic curve-based key agreement algorithm: 2n-bit keys

It is considered best practice to have key and data sizes match the desired cryptographic strength. However, in recognition that many devices will be extremely resource constrained, the guidelines below will discuss the minimum acceptable security levels.

[GDL 22] Cryptographic algorithms used in a protocol should be selected to have the same cryptographic strength.

4.1 RANDOM NUMBER GENERATION

The generation of keys and random numbers is required by many cryptographic algorithms and protocols.

[GDL 23] Compliant SOTA systems should generate all keys and random data using a true hardware number generator (TRNG) entropy source compliant with either AIS-31 [3] or the FIPS Special Publications 800-90B [16] and 800-90C [17].

[GDL 24] Compliant SOTA systems should perform post-processing of the TRNG output using a FIPS 140-2 certifiable deterministic random number generator (DRBG).

[GDL 25] The data output by the TRNG/DRBG should pass the statistical test suites from Dieharder [1] and NIST [13] for evaluating random number generators.

4.2 SYMMETRIC KEY ENCRYPTION ALGORITHM

Compliant SOTA systems should use a standard symmetric key encryption algorithm of sufficient strength.

[GDL 26] Compliant SOTA systems should use a standard encryption algorithm with strength equal to or greater than the Advanced Encryption Standard (AES) with 128-bit keys [11].

4.3 CRYPTOGRAPHIC HASH ALGORITHM

Compliant SOTA systems should use a standard cryptographic hash function of sufficient strength.

[GDL 27] Compliant SOTA systems should use a collision-resistant, preimage-resistant cryptographic hash function with strength equal to or greater than SHA-256 [1].

4.4 DIGITAL SIGNATURE ALGORITHM

Compliant SOTA systems should use a standard digital signature algorithm.

[GDL 28] Compliant SOTA systems should use a digital signature algorithm with strength equal to or greater than the elliptic curve digital signature algorithm (ECDSA) with 256-bit keys [11].

4.5 KEY AGREEMENT ALGORITHM

A session key may need to be established to perform a software update. Compliant SOTA systems should support a secure session key agreement protocols.

[GDL 29] Compliant SOTA systems should use a key agreement algorithm with strength equal to or greater than the elliptic curve Diffie-Hellman (ECDH) algorithm with 256-bit keys [7].

The basic Diffie-Hellman algorithm is vulnerable to man-in-the-middle attacks. This is usually addressed by augmenting the Diffie-Hellman algorithm with some form of authentication such as Menezes-Qu-Vanstone (MQV) [7] or password-authenticated key exchange (PAKE) [1].

[GDL 30] Compliant SOTA systems should use a standard authenticated key agreement algorithm to establish session keys.

4.6 DIGITAL CERTIFICATES

X.509 [4] is one of the most widely-used certificate formats. The power and flexibility of the certificate format can make them difficult to use in resource-constrained systems, however. SOTA software update system designers may instead want to use certificates containing a fixed set of fixed-length fields, but certain fields will likely be standard.

[GDL 31] Compliant SOTA systems certificates should contain the following fields (which are standard X.509 fields):

1. Version
2. Serial number
3. Issuer ID
4. Validity dates
5. Permissions
6. Subject public key info
7. Certificate signature info

Certificates are typically chained, with an unbroken signature chain leading back to the root certificate. Also, revocation of certificates and certificate authorities should also be supported.

[GDL 32] Compliant SOTA systems certificate chains should always verify the certificate signatures all the way to the root certificate.

[GDL 33] Compliant SOTA systems should verify that the current date is in the "Validity Date" range.

[GDL 34] Compliant SOTA systems should support remote revocation of all certificates except the root certificate.

[GDL 35] Compliant SOTA systems should support the revocation of any Certificate Authorities except the Root Authority.

4.7 NETWORK AND POINT-TO-POINT CRYPTOGRAPHY

The car communicates with the backend servers through the gateway and should always use a standard network security protocol such as TLS. In addition, software updates may also be encrypted and signed by the backend servers for authentication and decryption by the target device. In this case, the software update would be doubly encrypted and signed.

Point-to-point cryptography protects the software update during transit between devices. Whether point-to-point cryptography is used will depend on the importance of the software update, as well as the capabilities of the target device. It may be the case that a software update travels through several devices before reaching the final target device. An intermediate device may authenticate and decrypt the software if the target device in the car does not have the capability to securely store keys or perform cryptography. However, the use of point-to-point cryptography is encouraged wherever possible.

[GDL 36] Compliant SOTA systems should always use a network security protocol such as TLS for communications between the backend servers and the gateway.

[GDL 37] Whenever feasible, compliant SOTA systems should support point-to-point encryption and authentication between backend servers and target devices.

4.8 PASSWORDS

[GDL 38] The use of passwords as a single factor for authentication in compliant SOTA software update systems is discouraged, as they tend to be homogenous, hard to change, and easy to crack.

[GDL 39] Compliant SOTA software update systems should use multifactor authentication, such as biometrics or a hardware token, which provides an access code/key and requires a personal identification number (PIN) to use, to authenticate operators into the system.

5 KEY MANAGEMENT PLAN

One of the most important documents in any cryptographic system is the key management plan (KMP). This is a detailed document describing the complete lifecycle of all the keys in the system. Below is an outline of the information typically contained in a KMP.

[GDL 40] Compliant SOTA software update systems should have a detailed KMP covering the topics listed below.

5.1 LIST OF KEYS

Figure 2 below contains a list of types of key which may be present in a SOTA software update system, with a brief description of each.

SOTA Software Update Keys	
Key	**Description**
Network Keys	Keys used to secure network communications between backend servers and the gateway.
Root Keys	Root signing keys for the entire system. They are used to sign and verify certificates for the Operational Certificate Keys and Operational System Keys.
Operational Certificate Authentication Key	Public key pair used to sign and verify certificates for Device Public Keys. Its certificate is signed by a Root Key.
Operational Software Authentication Key	Public key pair used in the day-to-day operation to sign and verify software updates. Its certificate is signed by the Operational Certificate Authentication Key.
Operational Key Agreement Key	Public key pair used for key agreement with devices in the system. Its certificate is signed by the Operational Certificate Authentication Key.
Device Authentication Key	Public key pair stored in device to authenticate device to system. Its certificate is signed by the Operational Certificate Authentication Private Key.
Device Key Agreement Key	Public key pair stored in device for key agreement with operational servers. Its certificate is signed by the Operational Certificate Authentication Private Key.
Device Symmetric Key	Long-term symmetric encryption key stored in device.
Session Key(s)	Ephemeral symmetric key(s) valid for a single software update. Deleted immediately after use.

Figure 2: SOTA Software Update Keys

Depending on how the system is designed, it may require only a subset of these keys. Also, a single key may be used for multiple purposes. For example, to save storage space on the device, the Device Authentication Key Pair and Device Key Agreement Key Pair may be the same key pair.

[GDL 41] The KMP for a compliant SOTA software update system should contain a complete list of all the keys used in the system.

5.2 KEY AND RANDOM DATA GENERATION

Both keys and random data should be generated with high-quality random number generators. It should be emphasized that all random data required for cryptographic protocols be generated this way. Algorithms such as ECDSA are extremely brittle, and the long-term secret key can be recovered by an adversary if the ephemeral nonces are poorly generated.

[GDL 42] Compliant SOTA software update systems should generate all keys and random data used in cryptographic protocols using random number generators meeting guidelines [GDL 23], [GDL 24] and [GDL 25] in Section 4.1.

[GDL 43] The KMP for a compliant SOTA software update system should describe how all the keys in the system are generated.

5.3 STORAGE AND BACKUP

Keys will be stored differently based on their type, how they are used, and where they reside. For example, root keys will typically have the most stringent storage and backup requirements, while ephemeral session keys should be deleted once the session has ended.

[GDL 44] Compliant SOTA software update systems should store and back up all keys in a manner appropriate for each key type.

[GDL 45] The KMP for a compliant SOTA software update system should describe how all the keys in the system are stored and backed up.

5.4 KEY DISTRIBUTION

Many of the keys in the system will need to be distributed from where they are initially generated. For example, the public key of a root signing key needs to be distributed in a secure, authenticated manner to all devices authenticating certificates signed by the root authority.

[GDL 46] Compliant SOTA software update systems should distribute all keys in a secure, authenticated manner.

[GDL 47] The KMP for a compliant SOTA software update system should describe how all the keys in the system are protected during distribution.

5.5 USAGE

There are many different types of keys, used in many different types of situations and hardware. It is important to document how all the keys are used in the system to avoid using any of them in an inappropriate or insecure manner.

[GDL 48] The KMP for a compliant SOTA software update system should describe how all the keys in the system are used.

5.6 KEY AND CERTIFICATE UPDATES

Keys and certificates often have expiration dates and will need to be replaced. A robust system should have established procedures for updating expired keys and certificates. The system should also have established procedures if an expired key or certificate is used.

[GDL 49] Compliant SOTA software update systems should have established procedures for updating expired keys and certificates.

[GDL 50] Compliant SOTA software update systems should have established procedures for when an expired key or certificate is encountered.

[GDL 51] The KMP for a compliant SOTA software update system should describe which keys and certificates have expiration dates and need to be updated, as well as the procedures for doing so.

[GDL 52] The KMP for a compliant SOTA software update system should describe how the system behaves when an expired key or certificate is encountered.

5.7 KEY AND CERTIFICATE REVOCATION

In a large security system such as this one, it is likely that some of the keys and certificates will become compromised and need to be revoked and replaced. Any robust security system must be prepared for this eventuality and be able to respond appropriately.

[GDL 53] Compliant SOTA software update systems should have established procedures for revoking and replacing compromised keys and certificates.

[GDL 54] Compliant SOTA software update systems should have established procedures for when a revoked key or certificate is encountered.

[GDL 55] The KMP for a compliant SOTA software update system should describe how key and certificate revocation and replacement are handled.

[GDL 56] The KMP for a compliant SOTA software update system should describe how the system behaves when a revoked key or certificate is encountered.

6 SOTA SOFTWARE UPDATE GUIDELINES CHECKLIST

Below is a reference list of all the SOTA software update guidelines, with brief descriptions and links to where each guideline is specified. Reviewers of automotive SOTA software update systems can use the checklist to help evaluate the compliance of these systems with the guidelines in this document.

SOTA Software Update Guideline Checklist		
Guideline	**Description**	**Score**
[GDL 1]	Defines in-scope threats at high level	
Evaluation Comments		
[GDL 2]	Side-channel attacks and attacks requiring physical modification of the vehicle are out of scope	
Evaluation Comments		
[GDL 3]	Defines which tampering attacks are in scope	
Evaluation Comments		
[GDL 4]	High-level digital certificate recommendation for software updates	
Evaluation Comments		
[GDL 5]	High-level digital certificate recommendation for software updates	
Evaluation Comments		
[GDL 6]	Recommendation to sign after encryption	
Evaluation Comments		
[GDL 7]	Software with invalid signature is never installed	
Evaluation Comments		
[GDL 8]	Allowing only authorized entities to update software	
Evaluation Comments		

SOTA Software Update Guideline Checklist

Guideline	Description	Score
[GDL 9]	Recommendation for software updates to include versioning information to prevent rollbacks	
Evaluation Comments		
[GDL 10]	Recommendation to use TLS for all network connections	
Evaluation Comments		
[GDL 11]	Recommendation discouraging use of SSL	
Evaluation Comments		
[GDL 12]	Recommendation to perform host-name verification of server	
Evaluation Comments		
[GDL 13]	Recommendation to determine what information needs to be changed	
Evaluation Comments		
[GDL 14]	Recommendation to retain important information	
Evaluation Comments		
[GDL 15]	Deliver software to only authorized devices	
Evaluation Comments		
[GDL 16]	Recommendation to perform threat modelling of the system	
Evaluation Comments		
[GDL 17]	Recommendation to fail gracefully during DoS attacks	
Evaluation Comments		

SOTA Software Update Guideline Checklist		
Guideline	**Description**	**Score**
[GDL 18]	Recommendation to use secure boot wherever possible	
Evaluation Comments		
[GDL 19]	Recommendation to utilize anti-malware protection wherever possible	
Evaluation Comments		
[GDL 20]	Update process never runs concurrently with other processes	
Evaluation Comments		
[GDL 21]	Sensitive data and keys must be cleared after use	
Evaluation Comments		
[GDL 22]	Cryptographic algorithms used in a protocol should be selected to have the same cryptographic strength	
Evaluation Comments		
[GDL 23]	TRNG guideline	
Evaluation Comments		
[GDL 24]	TRNG guideline	
Evaluation Comments		
[GDL 25]	TRNG guideline	
Evaluation Comments		
[GDL 26]	Encryption algorithm guideline	
Evaluation Comments		

SOTA Software Update Guideline Checklist

Guideline	Description	Score
[GDL 27]	Cryptographic hash algorithm guideline	
Evaluation Comments		
[GDL 28]	Digital signature algorithm guideline	
Evaluation Comments		
[GDL 29]	Key agreement guideline	
Evaluation Comments		
[GDL 30]	Key agreement guideline	
Evaluation Comments		
[GDL 31]	Recommended fields for digital certificate	
Evaluation Comments		
[GDL 32]	Digital certificate verification guideline	
Evaluation Comments		
[GDL 33]	Recommendation to verify the before/after dates on each certificate	
Evaluation Comments		
[GDL 34]	Digital certificate revocation guideline	
Evaluation Comments		
[GDL 35]	Certificate Authority revocation guideline	
Evaluation Comments		

SOTA Software Update Guideline Checklist		
Guideline	**Description**	**Score**
[GDL 36]	Recommendation for network security between backend servers and the Gateway	
Evaluation Comments		
[GDL 37]	Recommendation to use point-to-point encryption whenever feasible	
Evaluation Comments		
[GDL 38]	Recommendation discouraging use of passwords	
Evaluation Comments		
[GDL 39]	Recommendation for compliant systems to use multifactor authentication	
Evaluation Comments		
[GDL 40]	Recommendation for compliant systems to have a detailed KMP	
Evaluation Comments		
[GDL 41]	Recommendation for KMP to contain complete list of keys in the system	
Evaluation Comments		
[GDL 42]	Recommendation to generate all keys using a high-quality random number generator	
Evaluation Comments		
[GDL 43]	Recommendation for KMP to describe generation of all the keys in the system	
Evaluation Comments		

SOTA Software Update Guideline Checklist

Guideline	Description	Score
[GDL 44]	Recommendation to store and back up all keys in an appropriate manner	
Evaluation Comments		
[GDL 45]	Recommendation for KMP to describe how all the keys in the system are stored and backed up	
Evaluation Comments		
[GDL 46]	Recommendation to distribute all keys in a secure, authenticated manner	
Evaluation Comments		
[GDL 47]	Recommendation for KMP to describe how all the keys in the system are distributed	
Evaluation Comments		
[GDL 48]	Recommendation for KMP to describe how all the keys in the system are used	
Evaluation Comments		
[GDL 49]	Recommendation to have established procedures for updating keys	
Evaluation Comments		
[GDL 50]	Recommendation to have established procedures if an expired key or certificate is encountered	
Evaluation Comments		
[GDL 51]	Recommendation for KMP to describe which keys and certificates expire and will need to be updated, as well as the procedures for doing so	
Evaluation Comments		

SOTA Software Update Guideline Checklist

Guideline	Description	Score
[GDL 52]	Recommendation for KMP to describe the behavior of the system if an expired key or certificate is encountered	
Evaluation Comments		
[GDL 53]	Recommendation to have established procedures for revoking and replacing keys and certificates	
Evaluation Comments		
[GDL 54]	Recommendation to have established procedures if a revoked key or certificate is encountered	
Evaluation Comments		
[GDL 55]	Recommendation for KMP to describe procedures for revoking and replacing keys and certificates	
Evaluation Comments		
[GDL 56]	Recommendation for KMP to describe the behavior of the system if a revoked key or certificate is encountered	
Evaluation Comments		

Figure 3: SOTA Software Update Guideline Checklist

7 REFERENCES

[1] M. Bellare, D. Pointcheval and P. Rogaway, Authenticated Key Exchange Secure against Dictionary Attacks, Advances in Cryptology – Eurocrypt 2000, LNCS, Springer-Verlag, Vol. 1807, pp. 139–155

[2] R. G. Brown, D. Ettelbuettel and D. Bauer, Dieharder: A Random Number Test Suite, Version 3.31.1, www.phy.duke.edu/~rgb/General/dieharder.php

[3] BSI, AIS-31: Evaluation of Random Number Generators, Version 0.10, 2013

[4] T. Dierks and E. Rescorla, IETF RFC 5246: The Transport Layer Security (TLS) Protocol, Version 1.2, August 2008

[5] A. Freier, P. Carlton, and P. Kocher, IETF RFC 6101: The Secure Sockets Layer (SSL) Protocol, Version 3.0, August 2011

[6] R. Housley, W. Polk, W. Ford, W., and D. Solo, IETF RFC 3280, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", April 2002

[7] IEEE Standard 1363-2000: IEEE Standard Specifications for Public-Key Cryptography, August 2000

[8] P. C. Kocher, J. Jaffee and B. Jun, Differential Power Analysis, Advances in Cryptology – Crypto'99, LNCS, Springer-Verlag, Vol. 1666, pp. 388-397

[9] NIST, FIPS 140-2, Security Requirements for Cryptographic Modules, December 2002

[10] NIST, FIPS 180-2, Secure Hash Standard, August 2002

[11] NIST, FIPS 186-4, Digital Signature Standard, July 2013

[12] NIST, FIPS 197, Advanced Encryption Standard (AES), November 2001

[13] NIST, Special Publication 800-22, Rev. 1a, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, April 2010

[14] NIST, Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation, December 2001

[15] NIST, Special Publication 800-90A Revision 1: Recommendation for Random Number Generation Using Deterministic Random Bit Generators, January 2015

[16] NIST, Special Publication 800-90B: Recommendation for the Entropy Sources Used for Random Bit Generation, August 2012

[17] NIST, Special Publication 800-90C: Recommendation for Random Bit Generator Constructions (Second Draft), April 2016

[18] B. S. E. Shoenfield, Securing Systems: Applied Security Architecture and Threat Models, ISBN-13 978-1482233971, CRC Press, 2015

8 COPYRIGHT AND OTHER LEGAL INFORMATION

This document is provided for informational purposes only, on an "AS-IS" basis. FASTR Inc. and its members and agents disclaim all liability arising from use of the information in this document; readers use the information at their own risk.

FASTR and FASTR's chevron logo are service marks of FASTR Inc. and may be used only with FASTR's permission.