Technical Article
10-2020

How to design FIPS 140-2 cryptographic modules to meet TCG Implicit Identity Based Device Attestation

Avi Avanindra (Infineon), Sergey Ostrikov (Infineon),
Travis Spann (Aegisolve)

This paper describes how to design and implement cryptographically secure devices, which meet the security requirements of FIPS 140-2[1] validation, while taking advantage of the Trusted Computing Group (TCG) defined device identification and attestation architecture.

FIPS 140-2 is a U.S. government computer security standard used to approve cryptographic modules. Secure systems depend on secret/private cryptographic keys. FIPS PUB 140-2 defines an entire category of requirements for the generation and management of these cryptographic keys, that must be met by the cryptographic module to be successfully validated by the National Institute of Standards and Technology (NIST). Some of those requirements pertain to the generation and use of random numbers for key management.

TCG has defined the Device Identifier Composition Engine (DICE) and Implicit Identity Based Device Attestation architecture to enable enhanced security and privacy for connected devices with minimal cost, both in hardware and software. Device attestation or certification is needed to know the cryptographic state of a system, to be able to trust it. The TCG DICE emerges as a very useful standard to ensure that a system uses authentic hardware and firmware.

The FIPS 140-2 requirements for generation and management of cryptographic keys, including the use of random numbers, and the TCG DICE/attestation architecture present a different set of requirements. These must be carefully aligned to meet both the standards and enable the DICE attestation architecture to be successfully validated by NIST for critical applications. This paper lays out the detailed problems and provides an approach to meet the requirements for both TCG and NIST/FIPS 140-2.

What is FIPS 140-2?

The Federal Information Processing Standard Publication 140-2 (FIPS PUB 140-2), is a U.S. government computer security standard used to approve cryptographic modules. FIPS 140-2 validation is required for any security product sold to the US government and its affiliated agencies, as well as in healthcare and financial industries to protect sensitive data. FIPS 140-3[2] supersedes the publication 140-2 and provides the latest set of security requirements for cryptographic modules; note that this paper does not describe FIPS 140-3 requirements which may be addressed in a subsequent revision to this paper in the future. With the growth in autonomous driving and related safety concerns, there is increasing demand to have FIPS 140-2 validation for devices used in autonomous vehicles. National Highway Traffic Safety Administration (NHTSA) has proposed that all Vehicle to Vehicle (V2V) equipment be “hardened” (FIPS-140 level 3) against intrusion by entities attempting to steal its security credentials[3]. The emerging Internet of Things (IoT) segment is driving the development of internet connected products, which demand a high level of security to be viable in the market. Security is playing an increasing role in product design, as more and more devices are Internet connected. FIPS 140-2 validation provides an assurance that the device meets the specified level of security. An increasing number of products are demanding security validations. Once validated, these products also receive premium pricing.

The FIPS 140-2 standard provides four increasing, qualitative levels of security: Level 1, Level 2, Level 3, and Level 4. The differences between the assessment levels relates to the sophistication of authentication methods, cryptographic key management, tamper resistance mechanisms, and the associated cryptographic module design assurance. These levels are intended to cover a wide range of potential applications and environments in which cryptographic modules may be employed. The Cryptographic Module Validation Program (CMVP) validates cryptographic modules to Federal Information Processing Standard FIPS PUB 140-2. Once successfully validated, the module is listed on CMVP[4] website.

The FIPS 140-2 standard specifies eleven categories of assertions that must be met and tests that must be passed to successfully validate a cryptographic module. One of these categories pertains to Cryptographic Key Management, and key generation is a critical part of the Cryptographic Key Management requirements.

FIPS 140-2 requirements for Key Generation and Entropy

Secure systems rely on secret/private cryptographic keys to provide the secure services of the cryptographic module. Approved Random Bit Generators (RBGs) are used to generate these keys. As per NIST Special Publication SP 800-133r2[5], “All keys shall be based directly or indirectly on the output of an approved Random Bit Generator.” An RBG appropriate for use in FIPS 140-2 validated cryptographic modules consists of two components:

1. A source of randomness, i.e. entropy source
2. A deterministic algorithm, i.e. deterministic random bit generator (DRBG) that takes an entropy input and uses it to produce pseudorandom values

The output of the DRBG is provided as an input to cryptographic functions for the purposes of key generation. The design and testing requirements of the entropy source are specified in NIST SP 800-90B[6]. The algorithms to accumulate these random bits – the DRBG are specified in NIST SP 800-90Ar1[7] Another specification SP 800-90C[8] provides the “glue” for putting the entropy source together with the algorithm to implement an RBG.

The following Figure 1 shows the construction of an Approved Random Number Generator as per NIST specifications:

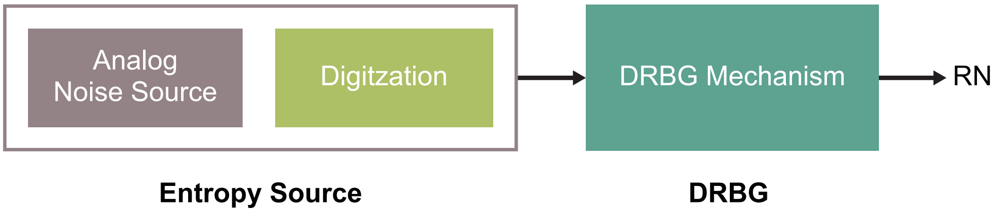

Figure 1: FIPS 140-2 Approved Random Number Generator

The DRBG is configured to accumulate a sufficient number of bits from the entropy source to meet the randomness requirement of a given application. It then converts these random bits into a bit string of a desired length to be used in cryptographic algorithms. Note the distinction of input bits from the entropy source into the DRBG to ensure randomness and the output of the DRBG which is a different bit string generated from those random bits.

Entropy Source

As shown in Figure 1, above, an entropy source usually consists of an analog noise source followed by digitization to generate random bit strings. Optionally, the digitized output can be post processed to accumulate the entropy, before being fed to the DRBG. The Entropy source used in the cryptographic module must be validated as per FIPS 140-2 Area 7 Cryptographic Key Management Requirements. In essence, the validation consists of generating one million samples of the raw data as well as one million samples of the restart data (e.g. one thousand restarts of the noise source with one thousand samples generated each time) and performing analysis, to estimate the entropy of the source. This is documented and reported to NIST as part of the FIPS 140-2 validation process.

Generic DRBG Functional Model as per SP 800 90Ar1

The DRBG Functional Model consists of an Instantiate function, Reseed function, Generate function and an Uninstantiate function. Instantiate acquires entropy input and may combine it with a nonce and a personalization string to create a seed from which the initial internal state is created. The Generate function uses the internal state to produce and output pseudo-random numbers, which is the essential purpose of a DRBG. A reseed acquires fresh entropy inputs and is required when the Generate function exceeds the maximum number of random numbers allowed using one seed value. Uninstantiate zeroizes (i.e destroys) the internal state. The Figure 2 below shows the DRBG Functional Model, as required by NIST SP 800-90Ar1 specification.

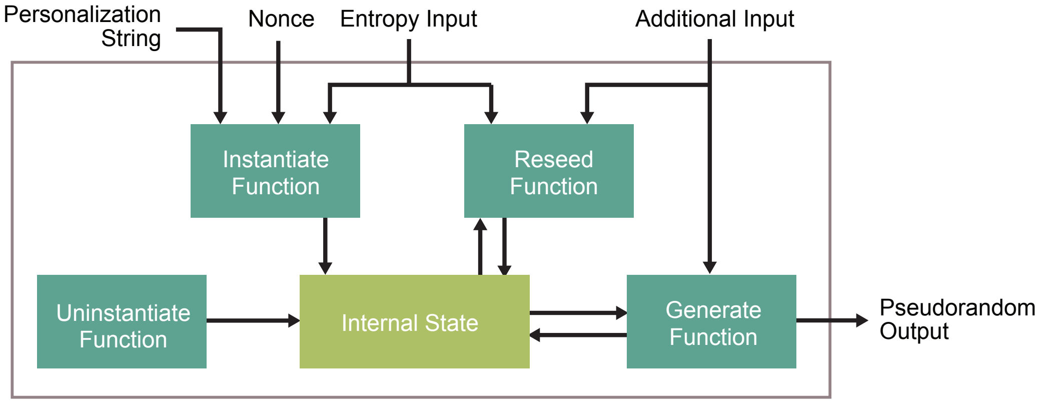

Figure 2: SP 800 90Ar1 DRBG Functional Model

The pseudorandom number generated from the output of a DRBG, can only be used for generation of cryptographic keys in a particular manner. Regeneration of the same DRBG output is not acceptable. This is a critical FIPS 140-2 requirement. If a new key is required, a new random number must be created by a call to the Generate function and, if needed, to the Reseed function (in case seed life has expired). Furthermore, as per SP800-90Ar1: “The seed that is used to initialize one instantiation of a DRBG shall not be intentionally used to reseed the same instantiation or used as the seed for another DRBG instantiation.”

FIPS based Design for Asymmetric Key Generation

With the above requirements in mind, the following diagram shows an implementation of the asymmetric key generation process for N public/private key pairs.

An entropy source is used to seed the DRBG. The instantiate function is then used to initialize the internal state of the DRBG. For each of the public/private key pairs, the generate function in DRBG is called to create a unique pseudorandom number (RNDx) that is provided as input to the Asymmetric Key Generation block in accordance with FIPS 186-4[9] for ECDSA key pairs and/or SP800-56Ar3[10] for ECC CDH key pairs.

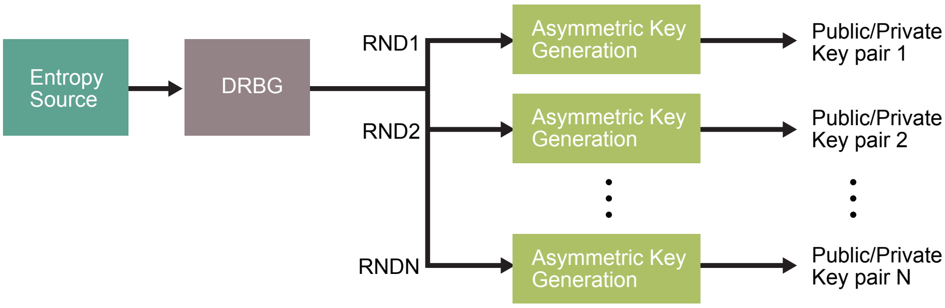
Entropy Source
DRBG
RND1
RND2
RNDN
Asymmetric Key Generation
Asymmetric Key Generation
Asymmetric Key Generation
Public/Private Key pair 1
Public/Private Key pair 2
Public/Private Key pair N

Figure 3: Asymmetric Key pair generation as per FIPS 140-2

So, what is the apparent misalignment with TCG DICE?

Trusted Computing Group (TCG) has defined the Device Identifier Composition Engine[11] (DICE) and Implicit Identity Based Device Attestation[12] architecture to enable enhanced security and privacy for connected devices with minimal cost, both in hardware and software. The purpose is to establish a strong device identity rooted in hardware and a mechanism to attest or certify that identity. The architecture binds the firmware running on this device to the Device Identity. Any attempts to change the firmware also changes the device identity and the attestation certificate. Figure 4 below depicts the TCG defined architecture for establishing a strong device identity and attestation.

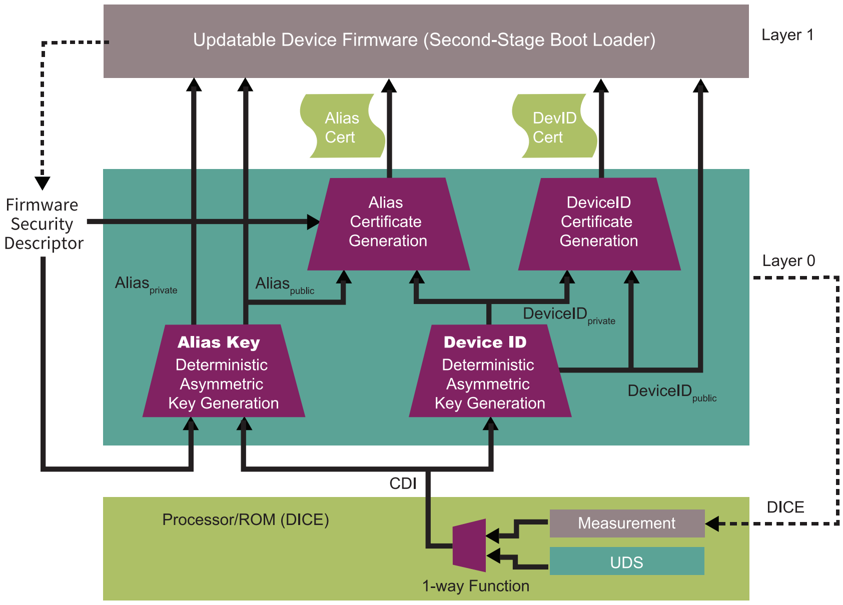
Updatable Device Firmware (Second-Stage Boot Loader)
Layer 1
Alias Cert
DevID Cert
Firmware Security Descriptor
Alias Certificate Generation
DeviceID Certificate Generation
Layer 0
Alias$_{private}$
Alias$_{public}$
DeviceID$_{private}$
Alias Key Deterministic Asymmetric Key Generation
Device ID Deterministic Asymmetric Key Generation
DeviceID$_{public}$
CDI
Processor/ROM (DICE)
DICE
Measurement
UDS
1-way Function

Figure 4: Implicit Identity Based Device Attestation Architecture

Unique Device Secret (UDS) is a statistically unique, cryptographic device secret and is the root of trust for this architecture. DICE measures the Layer 0 code and mixes it with the UDS to generate a Compound Device Identifier (CDI). CDI ensures that the device booted the first (Layer 0) code provided by the manufacturer. CDI is input to Layer 0 whose goal is to establish device identity and attestation. Layer 1 represents the Device firmware (FW) and other services it provides. The Firmware Security Descriptor (FSD) is a measure of the Layer 1 FW, that represents the identity of the Layer 1 FW. A cryptographic hash of the Layer 1 FW Image is an example of FSD that can be input to Layer 0. The DeviceID (DevID) is represented cryptographically as an asymmetric key pair, so the public key can be freely shared while keeping the private key a secret. Another layer of security is added by generating an Alias Key, also an asymmetric key pair, to certify the identity of the device. The Alias Key pair is generated using the CDI and a measure of the Layer 1 firmware. The DevID generated by Layer 0 is used to cryptographically sign the Alias Key certificate. The Alias Key pair, Alias certificate, and DevID certificate are available to device FW at Layer 1. The Alias Key pair and certificate can be used by the Device FW for authentication and attestation and provide other security services.

The Alias Key pair is generated every time the device powers up and the Layer 0 code is executed. If the Layer 1 code or CDI is changed, this will create a new Alias Key pair, changing the Alias public key and certificate which will prevent device attestation using the existing Alias public key. Hence, the identity represented by Alias Key needs to be unchanged as long as the Layer 1 device FW does not change (i.e. the Alias Key pair needs to be re-generated upon every boot, yet it needs to have the same value). This is a fundamental concept in DICE. The DICE specifies that a 1-way Function must be used for key generation. Typically, the term 1-way Function refers to a cryptographic hashing function.

Herein lie the apparent differences in the requirements between TCG DICE/Device Attestation and FIPS 140-2:

A) DICE requires the UDS to be bound to hardware which may be interpreted to mean that UDS must be hardmasked into the silicon ROM. Since the plaintext UDS is considered a FIPS 140-2 critical security parameter (CSP) it must be zeroizable. FIPS 140-2 zeroization entails an active overwriting of the memory that contains the plaintext CSP. It is not possible for a cryptographic module to zeroize values that are hardmasked into silicon ROM, thus hardmasking UDS into silicon ROM for purposes of DICE is non-compliant with FIPS 140-2.

B) FIPS 140-2 requires key generation to be performed using SP800-90Ar1 DRBG in conjunction with applicable supporting NIST Approved standards. Generating cryptographic keys directly via a 1-way hashing function is non-compliant with FIPS 140-2.

C) As per SP800-90Ar1: “The seed that is used to initialize one instantiation of a DRBG shall not be intentionally used to reseed the same instantiation or used as the seed for another DRBG instantiation”. Every time the device is powered up, a DRBG is instantiated, and it must use a new seed from the Entropy source. But this means that CDI will not remain the same and will change the device identity thus incompatible with DICE/Device Attestation architecture.

How to align FIPS 140-2 with TCG Implicit Identity Based Device Attestation Architecture

The following Figures 5 and 6 describe an implementation proposed by Infineon, compliant with FIPS 140-2 that also meets the TCG device attestation architecture. Figure 5 describes the operations at first boot up of the device when the Device Identity is established. This may be done at manufacturing or at customer site when provisioning the device. At first boot up, a validated entropy source generates the initial seed for input to DRBG. The DRBG is instantiated and its internal state is established. The DRBG generates a pseudorandom number RND1. This is the UDS in DICE architecture. The Generate function of DRBG is called two more times to generate pseudorandom numbers RND2 and RND3. For generation of RND2, the measure of Layer 0 code, as represented by SHA256 hash is used as an additional input to the DRBG. For generation of RND3, SHA256 hash of the Layer 1 FW image, representing the FSD is used as additional input to the DRBG. The RND1/UDS is input to a FIPS compliant Key Derivation Function (KDF), such as SP 800-108 KDF[13], with the SHA256 hash of Layer 0 as additional input. The additional input to KDF can be termed either context or label as defined in SP 800-108 KDF depending on the implementation. The KDF generates the CDI for use of Layer 0 as well as a DevID encryption/decryption key as shown in the Figure 5 below. The RND2 is directly passed to Layer 0 for the generation of DevID asymmetric key pair. The SHA256 hash of Layer 0 as an additional input to DRBG binds the Layer 0 FW and hardware rooted DICE to DevID key pair. For the generation of RND3 the FSD of Layer 1 is input to the DRBG to bind the FSD to the Alias Key, generated using RND3. The Layer 0 code generates the AliasPriv, AliasPub, DevIDPriv, DevIDPub keys as well as Alias and DevID certificates. The CDI from Layer 0 is input to another KDF that takes the FSD as an additional input and generates the Alias encryption key. The DevID and Alias encryption keys are used to encrypt and store the DevID and Alias Key pairs respectively. The control is then passed to Layer 1 device FW, which remains the same as in Figure 4 above.

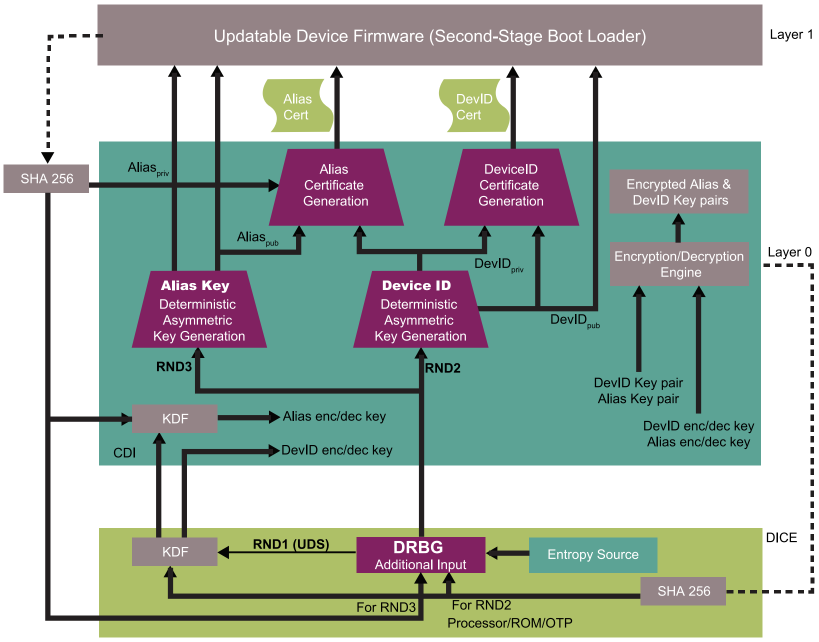

Figure 5: Implicit Identity Based Device Attestation Architecture – First Boot

Figure 6 below describes the operations that take place for all subsequent boot ups of the device. The DICE measures the Layer 0 code using SHA 256 hash, which is then input to KDF along with device UDS to generate the CDI. The KDF also generates the DevID encryption/decryption key. The layer 0 uses the CDI and FSD of Layer 1 as inputs to KDF to generate the Alias encryption/decryption key. These keys are then used to decrypt and authenticate the stored DevID and Alias Key pairs. An Approved encryption and authentication algorithm such as AES-GCM (SP 800-38D[14]), AES-CCM (SP 800-38C[15]) or Approved AES (SP 800-38A[16]) along with an Approved authentication mechanism such as HMAC (FIPS 198-1[17]) can be used for such purposes. Once the DevID and Alias Key pairs are successfully decrypted and authenticated, they are used to generate the Alias and DevID certificates as before. If the decryption and/or authentication fails, it implies the device FW either in Layer 0 or Layer 1 has been altered and hence valid certificates are not generated.

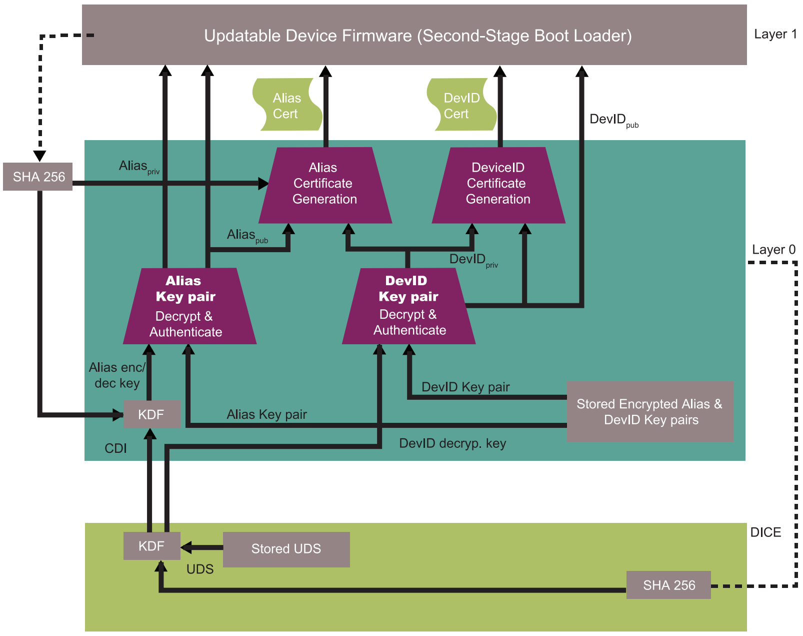
Updatable Device Firmware (Second-Stage Boot Loader)
Layer 1
Alias Cert
DevID Cert
DevID pub
SHA 256
Alias priv
Alias Certificate Generation
DeviceID Certificate Generation
Alias pub
DevID priv
Layer 0
Alias Key pair Decrypt & Authenticate
DevID Key pair Decrypt & Authenticate
Alias enc/ dec key
DevID Key pair
Stored Encrypted Alias & DevID Key pairs
KDF
Alias Key pair
CDI
DevID decryp. key
DICE
KDF
Stored UDS
UDS
SHA 256

Figure 6: Implicit Identity Based Device Attestation Architecture – Operational

A Layer 1 Device FW update process will call the DRBG with the new FSD as additional input, to generate a new pseudorandom number, that is used to create a new Alias Key pair as described in Figure 5. The new Alias Key pair is then stored encrypted/authenticated as before using a new Alias encryption/decryption key. The Alias encryption/decryption key is the output of Layer 0 KDF using the new FSD and existing CDI.

Since the Alias and DevID Key pairs are stored encrypted, FIPS 140-2 highly recommends that integrity checks be performed on the keys before use as per FIPS 140-2 IG 7.16[18]. Such recommendation is vacuously satisfied in this proposal via use of Approved authentication in conjunction with Approved encryption/decryption operations as described above.

Conclusion

In today's world of increasing connectivity, it is imperative that the privacy and security of the connected devices over the network is maintained. FIPS 140-2, a US government security validation certificate is a must have for devices sold to US government and for safety and security critical private industries. TCG has defined an architecture of robust security features through their DICE and implicit Identity Based Device Attestation Specifications. This paper describes how compliance with both FIPS-140 and TCG DICE is possible. FIPS 140-2 requires that the cryptographic keys be generated from the output of a DRBG, and that a DRBG output be used only once to generate a cryptographic key. DICE requires the Alias Key pairs, that represent the device identity to be regenerated by the layer 0 code every time the device boots up. This is not necessary, as demonstrated in this paper. A solution has been presented that allows a device to be TCG DICE compatible without relaxing any FIPS 140-2 validation requirements.

References

1. FIPS PUB 140-2
https://csrc.nist.gov/publications/detail/fips/140/2/final
2. FIPS PUB 140-3
https://csrc.nist.gov/publications/detail/fips/140/3/final
3. NHTSA V2V Proposal
https://www.federalregister.gov/documents/2017/01/12/2016-31059/federal-motor-vehicle-safety-standards-v2v-communications
4. CMVP Website
https://csrc.nist.gov/projects/cryptographic-module-validation-program
5. SP 800-133r2: Recommendation for Cryptographic Key Generation
https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-133r2.pdf
6. SP 800-90 B: Recommendation for the Entropy Sources Used for Random Bit Generation
https://csrc.nist.gov/publications/detail/sp/800-90b/final
7. SP 800-90 A rev 1: Recommendation for Random Number Generation Using Deterministic Random Bit Generators
https://csrc.nist.gov/publications/detail/sp/800-90a/rev-1/final
8. SP 800-90C: Recommendation for Random Bit Generator (RBG) Constructions
https://csrc.nist.gov/publications/detail/sp/800-90c/draft
9. FIPS 186-4: Digital Signature Standard (DSS)
https://csrc.nist.gov/publications/detail/fips/186/4/final
10. SP 800-56Ar3: Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography
https://csrc.nist.gov/publications/detail/sp/800-56a/rev-3/final
11. Hardware Requirements for a Device Identifier Composition Engine
https://trustedcomputinggroup.org/resource/hardware-requirements-for-a-device-identifier-composition-engine/
12. Implicit Identity Based Device Attestation
https://trustedcomputinggroup.org/resource/implicit-identity-based-device-attestation/

13. SP800-108 KDF: Recommendation for Key Derivation Using Pseudorandom Functions
https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-108.pdf
14. SP 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC
https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf
15. SP 800-38C: Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality
https://csrc.nist.gov/publications/detail/sp/800-38c/final
16. SP 800-38A: Recommendation for Block Cipher Modes of Operation: Methods and Techniques
https://csrc.nist.gov/publications/detail/sp/800-38a/final
17. FIPS 198-1: The Keyed-Hash Message Authentication Code (HMAC)
https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.198-1.pdf
18. FIPS 140-2 IG: Implementation Guidance for FIPS 140-2 and the Cryptographic Module Validation Program
https://csrc.nist.gov/csrc/media/projects/cryptographic-module-validation-program/documents/fips140-2/fips1402ig.pdf

Notice: The information provided within this paper is for general informational purposes only. While we try to keep the information up-to-date and correct, there are no representations or warranties, express or implied, about the completeness, accuracy, reliability, suitability or availability with respect to the information, products, services, or related graphics contained in this paper for any purpose.

About Infineon: Infineon Technologies AG is a world leader in semiconductor solutions that make life easier, safer and greener. Microelectronics from Infineon is the key to a better future. In the 2019 fiscal year (ending 30 September), the Company reported sales of €8.0 billion with around 41.400 employees worldwide. With the acquisition of US-based Cypress Semiconductor Corporation in April 2020, Infineon has become a global top 10 semiconductor company. Further information is available at www.infineon.com

About Aegisolve: Headquartered in Silicon Valley, California, AEGISOLVE is a trusted, independent, third-party cybersecurity laboratory accredited for FIPS cryptographic and security testing (NVLAP Lab Code: 200802-0). AEGISOLVE is the industry leader in providing Federal Information Processing Standards testing and validation certificates for many industries worldwide including IoT, automotive, cloud, banking, healthcare, telecommunications, critical infrastructure and digital cinema.

Published by
Infineon Technologies AG
81726 Munich, Germany

Document number:B000-I0000-V1-X-7600-EU-EC
Date:10/2020

Please note!
This Document is for information purposes only and any information given herein shall in no event be regarded as a warranty, guarantee or description of any functionality, conditions and/or quality of our products or any suitability for a particular purpose. With regard to the technical specifications of our products, we kindly ask you to refer to the relevant product data sheets provided by us. Our customers and their technical departments are required to evaluate the suitability of our products for the intended application.

We reserve the right to change this document and/or the information given herein at any time.

Additional information
For further information on technologies, our products, the application of our products, delivery terms and conditions and/or prices, please contact your nearest Infineon Technologies office (www.infineon.com).

Warnings
Due to technical requirements, our products may contain dangerous substances. For information on the types in question, please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by us in a written document signed by authorized representatives of Infineon Technologies, our products may not be used in any life-endangering applications, including but not limited to medical, nuclear, military, life-critical or any other applications where a failure of the product or any consequences of the use thereof can result in personal injury.