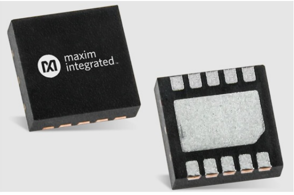
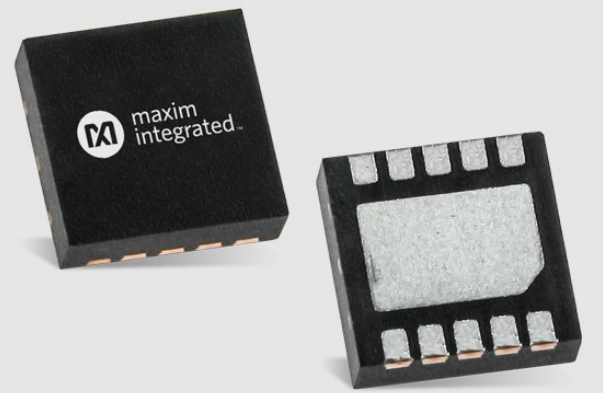
ANALOG DEVICES DS28C40 DeepCover Automotive I2C Authenticator



ANALOG DEVICES DS28C40 DeepCover Automotive I2C Authenticator User Manual

Contents

ANALOG DEVICES DS28C40 DeepCover Automotive I2C Authenticator



Introduction

The automotive industry is experiencing a transformation in the number of new features available to customers, on the road to having autonomous vehicles in every driver’s garage. These features contribute to conditional automation of steering, acceleration, and braking. Simultaneously, convenience and service features are available with the touch of a screen or through a voice command. Today, a car possesses the smarts to do much more on its own, from automatically switching on high beams to parking itself, detecting blind spots, and pre-emptive braking to avoid a collision. Making these capabilities possible are electronic control units (ECUs) that connect to the electronics of individual car parts. These electronics support items such as advanced driver assistance system (ADAS), power management, electric vehicle (EV) powertrain, infotainment, LED lighting, body electronics, mobile connectivity, and security to name a few. Many of the car parts require strict adherence to original equipment manufacturer (OEM) specifications to guarantee performance and safe operation. But how does the ECU know if these requirements are met?

This application note discusses how cryptographic mutual authentication schemes can enable a reliable authentication of the vital car parts through a pairing operation, helping to meet the ISO21434 and the UNECE WP.29 Cybersecurity Regulation requirements which enforce a "secure by design" paradigm.
Let us define what is meant by "pairing" in this context. Pairing is a cryptographic authentication and association between different vehicle subsystems that enables mutual trust. When it comes to the automotive edge, including automotive sensors and actuators, trust encompasses several aspects: car parts must be OEM-approved, provable, and have securely controlled life cycle (manufacturing, installation, calibration, refurbishing, decommissioning, etc.). The DS28C40 Automotive I2C Authenticator IC can support various pairing schemes and the traceability of the life cycle for the automotive edge. This IC is used as an example throughout this document.

Benefits of Pairing

- The use of car part pairing brings many benefits to automotive manufacturers. Let us describe these benefits in more detail.
- Identification and Strong Authentication of Parts
- The first and foremost benefit of "pairing" is to provide a cryptographically strong identification and authentication of car parts. By securely tying a specific car part to a specific vehicle, manufacturers can ensure that only authorized parts are used in their vehicles. This not only improves safety, but also helps to prevent fraud, theft, and counterfeiting. Risks are mitigated through this strong authentication scheme since any replacement part must now be authentic and valid, eliminating counterfeit or stolen parts.

Strong Authentication of Life Cycle Data

The second benefit of "pairing" is the ability to store and attest to the life cycle of a car part. This includes the part's calibration and settings, the life cycle state (manufacturing steps, maintenance steps, mounting and transferring to another car, configuring/calibrating, decommissioning, etc.), the associated car chassis identifier, and other relevant traceability information. Cryptographic methods using digital signature bring a formal proof of the full car part's state. This additional information can be used by the car's ECUs to manage otherwise authentic parts, like reject an OEM ADAS camera improperly calibrated, decommissioned, or voluntarily mounted into another car without proper authorization. This data can also be encrypted for added security, allowing manufacturers to ensure that only authorized parties can access it. This attestation of a part's life cycle reduces the risk of using invalid parts even though they are authentic, provided that the ECU is also secure enough so that no bypass of the car part verification is possible.

Secure Life Cycle Data Write Access Control

Being able to trust secure life, cycle data is vital to the trust. Tampering with the life cycle information could allow someone to refurbish otherwise worn-out or malfunctioning parts that would cause safety risks or use stolen parts. By using a cryptographic-based access control, manufacturers can ensure that only authorized parties can modify the car part's life cycle information memory and other information used to bind the car part to an ECU. An approved OEM dealer can then replace a car part and associate the part with the car chassis, run an approved calibration, etc.

Cryptographic Activation of Car Parts

There are multiple scenarios where a car part should cease functioning normally if it is not attached to a legit vehicle. This can occur when a part is stolen. Some devices need a proper installation process to operate safely. Being moved to another vehicle without following strict maintenance rules can be dangerous. The issue can also arise when a man-in-the-middle attack occurs, where the vehicle subsystem is not communicating directly with a legit ECU but with an intermediate rogue device. Cryptographic activation of a car part can solve those issues.

DS28C40 Key Features

The DS28C40 can be installed into a car part as to establish pairing. So, it is important to highlight the major features of DS28C40 before going into more details in other sections. The device is a secure authenticator that provides a core set of cryptographic tools. These tools provide symmetric and asymmetric security functions as highlighted in Table 1.

Table 1. DS28C40 Crypto-Security Type Comparison

CRYPTO-SECURITY TYPE	**DESCRIPTION**
Symmetric (SHA-256based)	The SHA-256 secret-key capabilities are compliant with FIPS 180 and are flexibly used either in conjunction with ECDSA operations or independently for multiple HMAC functions. Typically, it has the following implications for the system: 1. ECU and DS28C40 operate from the same or derived secret key. 2. Secret is protected from disclosure attack. 3. Supports bidirectional authentication by comparing HMACs with read/write of pages. 4. SHA-256 has lower algorithm complexity vs. ECC-P256. 5. SHA-256, when used, has up to 12x quicker computation time vs. ECC-P256.
Asymmetric (ECC-P256-based)	The ECC public/private key capabilities operate from the NIST-defined P-256 curve and include FIPS 186-4-compliant ECDSA signature generation and verification to support a bidirectional asymmetric key authentication model. Typically, it has the following implications for the system: 1. ECU operates with public key; DS28C40 with corresponding private key. 2. Private key must be protected; no requirement to protect public key. 3. Supports bidirectional authentication of signatures with read/write of pages. 4. ECC-P256 increased complexity vs. SHA-256. 5. ECC-P256 has longer authentication time vs. SHA-256.

In addition, the device contains an I2C interface, a true random number generator (TRNG), 6kb of one-time programmable (OTP) memory for user data, keys and certifications, one configurable general-purpose input/output (GPIO), and a unique 64-bit ROM identification number (ROMID). The OTP memory can only set bits from 1 to 0 in 32-byte memory pages. Protection settings exist for blocks of memory pages. With an OTP device, write operations and protection settings produce irreversible results. Protection settings are write/read protect, authenticated write protect for ECDSA/HMAC and more complex encrypted protections. The GPIO pin supports authenticated configurability. Lastly, the device fits in a 10-Pin TDFN (3mm x 3mm) package with operation range from -40°C to 125°C.

Different Key Installation Schemes Enable Various Car-to-Part Pairing Options

Benefits of pairing security mechanisms rely on a challenge-response authentication scheme that requires the installation of various “credentials”: certificates, public-private key pairs, static public keys, shared secret keys, etc. Various options provide different levels of security and flexibility, allowing manufacturers to choose the best option for their specific needs. Multiple options can be combined: in general, an initial vehicle part authentication is required to exclude counterfeits, but going beyond, additional steps including installation, configuration, and eventually, a specific vehicle-to-part association must be performed. These are covered in the remainder of the document.

Challenge-Response Authentication

As a general reminder, a challenge-response authentication has the following steps:

1. The verifier device sends a “challenge” (a random number) to prover device.
2. The prover uses a secret to digitally sign the random number and produce a “response” (the random number may be concatenated to attach additional information, e.g., life cycle data). Challenge-Response Authentication As a general reminder, a challenge-response authentication has the following steps:
    1. The verifier device sends a “challenge” (a random number) to prover device.
    2. The prover uses a secret to digitally sign the random number and produce a “response” (the random number may be concatenated to attach additional information, e.g., life cycle data).
    3. The prover sends the “response” to the ECU, sometimes with additional identification data.
    4. The verifier verifies the prover’s identification data and response.
    Note this authentication can be done both ways.

Certificate-Based Option

Car parts strong authentication, life cycle data authentication, memory access control, and car parts activation can rely on certificate-based authentication. So, this option cryptographically proves a device’s identity and additional information such as life cycle data.

As a general reminder, a certificate-based authentication of a car part by an ECU is as follows:

1. The ECU sends a “challenge” (a random number) to the attached car part.
2. The car part uses its private key to digitally sign the random number (the random number may be concatenated to attach additional car part information, e.g., life cycle data).
3. The car part sends the “response” signature and its car part certificate to the ECU.
4. The ECU verifies the car part’s certificate using the OEM CA certificate. Nonauthentic car parts get rejected here as their certificates are not issued by the OEM CA.
5. The ECU verifies the car part’s response. All clones are rejected since the private keys cannot be copied; therefore, cloned parts cannot calculate correct signatures. This scheme can also be performed in the opposite direction, where the car part can ensure that the ECU is legit, or a memory write access is granted.

Certificate Installation

Certificate installation is to be performed prior to running any certificate-based authentication scenarios. So, during the manufacturing of the car, public key certificates must be programmed into both the car’s ECU and the car parts attached to it. The installation of those certificates requires a mutually trusted root certification authority (OEM CA). Let us take the example of a car’s camera to be the associated car part to the ECU as per Figure 1.

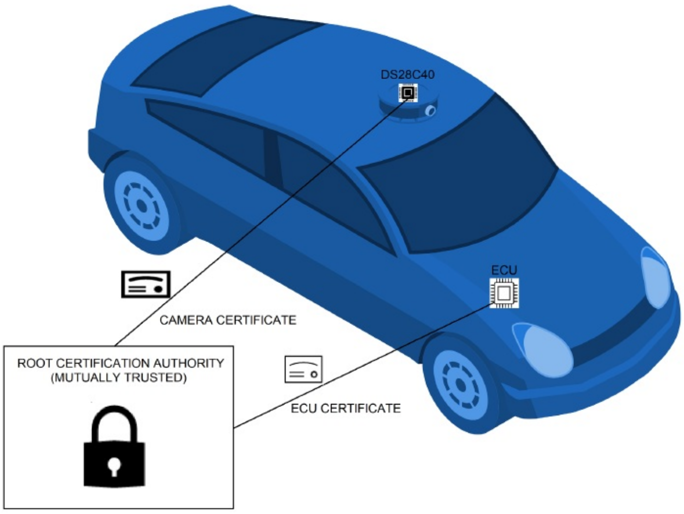


Figure 1. OEM CA Private Key issues all certificates.

ECU Programming

When the ECU needs to authenticate itself to vehicle's parts and their cryptographic activation is needed, the OEM must use a secure initialization system to prepare the ECU by performing the following operations:

- Load a key pair unique to the ECU and store it into the ECU (alternatively the key pair can be generated onboard the ECU, and the public key read out from the ECU). The private key helps the ECU prove its identity.
- Issue an ECU Certificate and store it into the ECU. The ECU Certificate is digitally signed using the OEM CA private key. This step proves that the ECU is from the OEM.
- Store the OEM CA Certificate (only the CA public key is stored to be used as a simple CA certificate) into the ECU. This step helps the ECU verify attached car part's certificates during strong authentication processes and make sure that they are approved OEM parts. An example of ECU Certificate is defined in Table 2.

Table 2. ECU's Certificate Content

FIELD	DESCRIPTION
ECU Unique Identifier	Optional. An ID uniquely identifying the ECU.
ECDSA Public Key X coordinate	The ECU has a unique key pair for authentication purposes. This field contains the public part of this key pair. As anyone can generate a pair of keys, certification is enacted on this public key to prove that it has been approved by a mutually trusted Root CA.
ECDSA Public Key Y coordinate	
Additional data	Additional arbitrary data
ECDSA Signature R component	This signature is calculated using the Root CA Authority private key when the certificate is “issued.” It seals the fields mentioned above; nobody can forge a certificate or modify a certificate. The certificates guarantee that the ECDSA Public Key value in the field above matches the ECU Unique identifier and arbitrary data claimed in the certificate AND has been issued by the intended Root CA. Valid signatures can be generated only by the legitimate Root CA.
ECDSA Signature S component	

The generation of the ECU’s certificate involves three entities that include the OEM CA, a secure initialization system, and the ECU as per Figure 2. The OEM CA contains a private key to sign all ECU certificates. It must obviously be protected against disclosure to avoid generation of rogue ECUs and car parts. Typically, it is contained in a secure facility. The initialization system is really the tool that carries out the procedure as illustrated in Figure 2. The manufacturer of the ECU or car company should develop this tool.

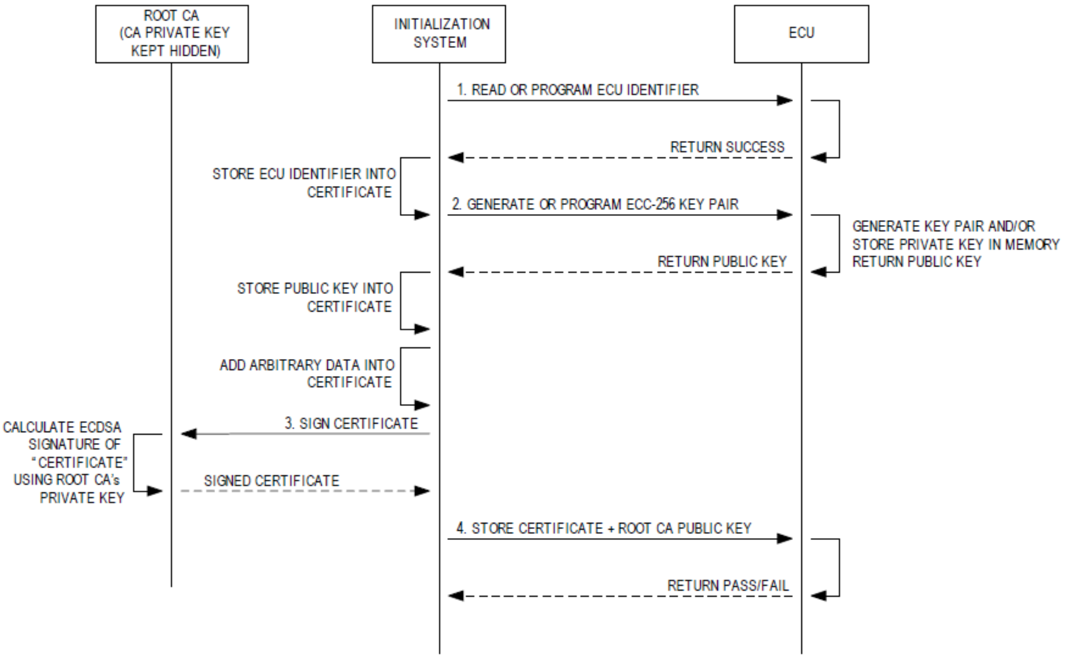

ROOT CA (CA PRIVATE KEY KEPT HIDDEN)
INITIALIZATION SYSTEM
ECU
1. READ OR PROGRAM ECU IDENTIFIER
RETURN SUCCESS
STORE ECU IDENTIFIER INTO CERTIFICATE
2. GENERATE OR PROGRAM ECC-256 KEY PAIR
GENERATE KEY PAIR AND/OR STORE PRIVATE KEY IN MEMORY RETURN PUBLIC KEY
RETURN PUBLIC KEY
STORE PUBLIC KEY INTO CERTIFICATE
ADD ARBITRARY DATA INTO CERTIFICATE
3. SIGN CERTIFICATE
CALCULATE ECDSA SIGNATURE OF “CERTIFICATE” USING ROOT CA's PRIVATE KEY
SIGNED CERTIFICATE
4. STORE CERTIFICATE + ROOT CA PUBLIC KEY
RETURN PASS/FAIL

Figure 2. Generation of the ECU's certificate.

Car Part Programming

Car parts must also undergo a similar initialization process so they can be authenticated by the ECU. The OEM CA, with the assistance of an initialization system, must generate a pair of keys unique to each device, then issue and load the car part’s certificate (e.g., camera), and eventually load the mutually trusted OEM Certificate into the car part. In this document, the DS28C40 IC is used to support this. The certificate format supported by the DS28C40 is shown in Table 3.

Table 3. Camera’s Certificate Content, Stored in DS28C40

FIELD	**DESCRIPTION**
DS28C40 ROMID	Each DS28C40 has a 64-bit unique identifier called a ROMID.
ECDSA Public Key X coordinate	Each DS28C40 has a unique static key pair for authentication purposes. This field contains the public part of this key pair. As anyone can generate a pair of keys, certification is enacted on this public key to prove that it has been approved by a mutually trusted Root CA.
ECDSA Public Key Y coordinate	
Additional data	Additional arbitrary data
ECDSA Signature R component	This signature is calculated using the Root CA Authority private key when the certificate is “issued.” It seals the fields mentioned above; nobody can forge a certificate or modify a certificate. The certificates guarantee that the ECDSA Public Key value in the field above matches the ROMID identifier and arbitrary data claimed in the certificate AND has been issued by the intended Root CA. Valid signatures can be generated only by the legitimate Root CA.
ECDSA Signature S component	

The camera’s certificate installation also involves three entities that are the CA, an initialization system, and the DS28C40 as per Figure 3. The CA possesses a private key for signing certificates. It must obviously be protected against disclosure to avoid generation of rogue car parts. Typically, it is contained in a secure facility. The initialization system is really the tool that carries out the procedure as illustrated in Figure 3.

In a first step, the DS28C40 self-generates a random public-private key pair within its secure memory, and outputs the public key to be certified by the CA. The CA then reads out the DS28C40 public key, generates a device certificate, and stores it back into the DS28C40 memory. Last, the initialization system may store the CA Certificate into the DS28C40 using a very simple format (only the public key of the CA, known as the Authority Public Key, is stored) By doing so, the car part can be cryptographically activated or the write access control to its memory can be enabled.

Very often, car parts are initialized by OEM-approved car part manufacturers, not the OEM itself. Certification schemes allow to delegate this step without the need for the OEM to share the security critical OEM CA private key. Car part manufacturers usually have their own CA, with their own certificate signing private key and CA Certificate. Their parts are accepted by the car ECUs if these hold the car part manufacturer’s CA Certificate. Therefore, the OEM oversees loading the right car part manufacturer certificates into ECUs so that the part manufacturer’s devices are accepted as “authentic.” The next figure depicts the initialization of a car part using the DS28C40.

If any secure writes are needed, then a Write Authority Public Key should be installed too (see Figure 3). This write authority public key is part of a second Root CA key pair only used for writing. Therefore, the Root CA contains another private key called the Write Authority Private Key as to distinguish its usage.

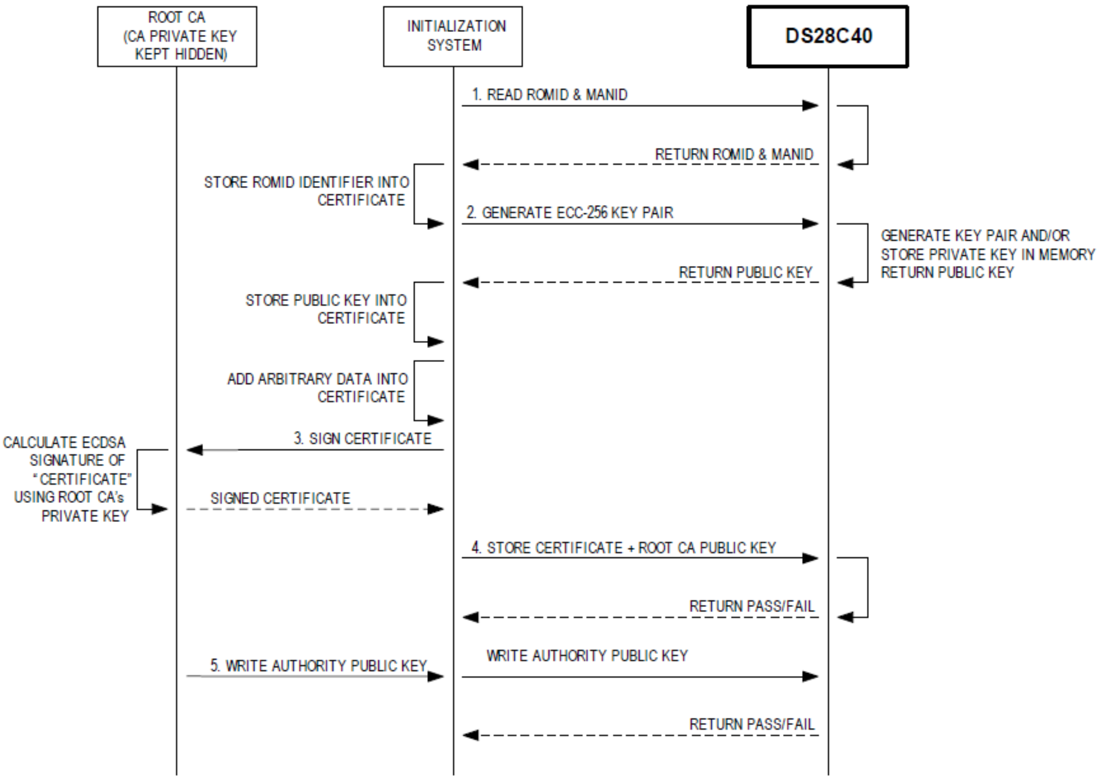

ROOT CA (CA PRIVATE KEY KEPT HIDDEN)
INITIALIZATION SYSTEM
DS28C40
1. READ ROMID & MANID
RETURN ROMID & MANID
STORE ROMID IDENTIFIER INTO CERTIFICATE
2. GENERATE ECC-256 KEY PAIR
GENERATE KEY PAIR AND/OR STORE PRIVATE KEY IN MEMORY RETURN PUBLIC KEY
RETURN PUBLIC KEY
STORE PUBLIC KEY INTO CERTIFICATE
ADD ARBITRARY DATA INTO CERTIFICATE
3. SIGN CERTIFICATE
CALCULATE ECDSA SIGNATURE OF "CERTIFICATE" USING ROOT CA's PRIVATE KEY
SIGNED CERTIFICATE
4. STORE CERTIFICATE + ROOT CA PUBLIC KEY
RETURN PASS/FAIL
5. WRITE AUTHORITY PUBLIC KEY
WRITE AUTHORITY PUBLIC KEY
RETURN PASS/FAIL

Figure 3. Generation of the DS28C40's Camera Certificate.

Additionally, Analog Devices, Inc.'s preprogramming service can perform steps revealed in Figure 3. It does this by ways of secure importation of the CA private key into Analog Devices' secure test facility. By doing so, it can make the generation of DS28C40 key pairs and certificates preprogrammed much easier for the car OEM.

Pairing Operation

When such a car part is mounted and attached to an ECU, an initial authentication must be performed by the ECU to ensure the part is from an approved part manufacturer thanks to the certificate-based scheme. This authentication is successful if the ECU can successfully verify the car part's response to the challenge. However, this does not protect against swapping parts without control as all parts issued by the same CA are valid. To bind the part further permanently to a specific car chassis, additional checks can be performed by the ECU using information stored within the car part's access-controlled memory. The information includes the car part's unique identifier and public key. For example, certificate-based pairing can be simply storing of unique identifiers or public keys between the ECU and car part.

Note that the whole authentication flow explained in this section must be performed every time (verify the certificate first and then the part's response) which represents a timing latency hit.

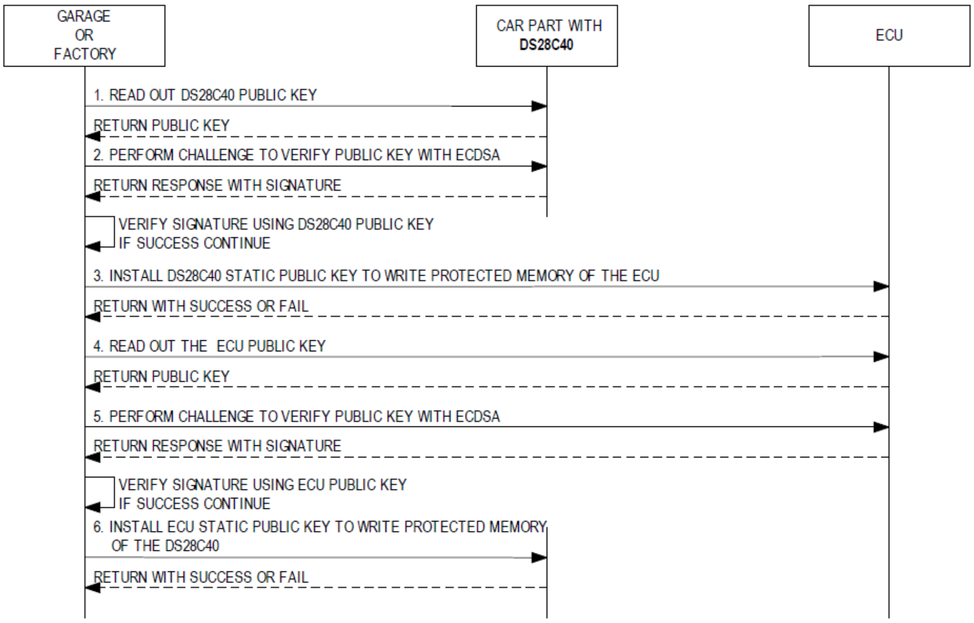

GARAGE
OR
FACTORY
CAR PART WITH
DS28C40
ECU
1. READ OUT DS28C40 PUBLIC KEY
RETURN PUBLIC KEY
2. PERFORM CHALLENGE TO VERIFY PUBLIC KEY WITH ECDSA
RETURN RESPONSE WITH SIGNATURE
VERIFY SIGNATURE USING DS28C40 PUBLIC KEY
IF SUCCESS CONTINUE
3. INSTALL DS28C40 STATIC PUBLIC KEY TO WRITE PROTECTED MEMORY OF THE ECU
RETURN WITH SUCCESS OR FAIL
4. READ OUT THE ECU PUBLIC KEY
RETURN PUBLIC KEY
5. PERFORM CHALLENGE TO VERIFY PUBLIC KEY WITH ECDSA
RETURN RESPONSE WITH SIGNATURE
VERIFY SIGNATURE USING ECU PUBLIC KEY
IF SUCCESS CONTINUE
6. INSTALL ECU STATIC PUBLIC KEY TO WRITE PROTECTED MEMORY
OF THE DS28C40
RETURN WITH SUCCESS OR FAIL

Figure 4. Static Public Key pairing of the DS28C40 and ECU.

Static Public Key Option

If the flexibility of the previous certification scheme is not needed, a simpler way of pairing can be achieved using static public keys. This can be done in a trusted location, during manufacturing or at a garage, for example, when installing/fixing a car's camera. Figure 4 shows the pairing of the car part with the ECU. In this scheme, the ECU's public key is directly stored into the car part's DS28C40 by an accredited actor, with the assumption that the public key is trustworthy. The reverse operation is also performed where the car part's (DS28C40) public key is directly written into the ECU memory, assuming it is trusted. Assuming the operation is run by a trusted party, the ECU and the part are now paired as they mutually trust each other's public key. If the part gets replaced by a new one, the latter is rejected by the ECU as the in-memory public key used by the ECU to verify the part's response does not match the new part's private key.

This scheme is simpler and faster to execute in that it excludes the issuance and verification of certificates, but it requires more trust in the part origin and pairing process as public keys are directly manipulated. If public keys can be controlled by a rogue player, this one can freely replace the mutually exchanged public keys and associate a new part. Note that this scheme skips the initial strong authentication of the car parts; therefore, counterfeits are not prevented.

Direct Shared Secret Key Installation

Pairing can also be done by directly preprogramming shared secret keys into both the ECU and the vehicle's parts. The benefit of this method is that it is the simplest and typically the quickest for cryptography computation calculations. Similar to the mutual exchange of public keys by a trusted actor mentioned above, a randomly chosen shared secret is directly stored both into the car part's DS28C40 and the ECU by an accredited actor. Assuming the operation was run by a trusted party, the ECU and the part are now paired as they shared the same secret key. If the part gets replaced by a new one, the correct secret key must be installed into the part; otherwise, it is rejected by the ECU as the in-memory shared key used by the ECU to verify that the part's response does not match the new part.

The DS28C40 can be delivered with preloaded shared secrets derived from a root secret key and a unique 64-bit ID combined with a 16-bit manufacturing ID. The root secret can then be directly stored into ECUs so each one of them can accept all car parts. Having the root secret allows derivation of the same shared key by the ECU. Alternatively, and to mitigate the risks inherent to the disclosure of the unique root key present in all ECUs, the trusted garage/manufacturer can directly load the vehicle part's shared secret into the ECU as per Figure 5.

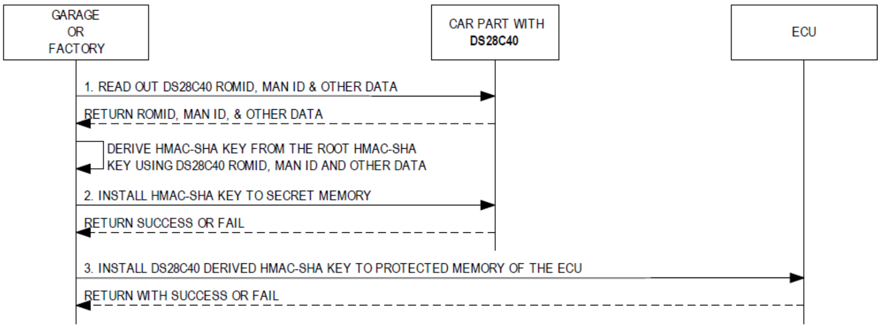

GARAGE
OR
FACTORY
CAR PART WITH
DS28C40
ECU
1. READ OUT DS28C40 ROMID, MAN ID & OTHER DATA
RETURN ROMID, MAN ID, & OTHER DATA
DERIVE HMAC-SHA KEY FROM THE ROOT HMAC-SHA
KEY USING DS28C40 ROMID, MAN ID AND OTHER DATA
2. INSTALL HMAC-SHA KEY TO SECRET MEMORY
RETURN SUCCESS OR FAIL
3. INSTALL DS28C40 DERIVED HMAC-SHA KEY TO PROTECTED MEMORY OF THE ECU
RETURN WITH SUCCESS OR FAIL

Figure 5. Pairing a shared key to both the DS28C40 and ECU.

Key Establishment Option

A hybrid option involves both an initial certificate-based authentication and a secret key-based pairing using the Elliptic-curve Diffie-Hellman (ECDH) protocol. The secret key-based pairing option requires the same steps as the certificate-based option. An additional step (ECDH) securely establishes a shared secret key between the ECU and the car part without revealing any sensitive information during pairing.

The benefits of key establishment option are multiple. The scheme easily allows to enable multiple part providers to produce OEM-certified parts: At installation time, parts can be fully authenticated as genuine. The additional ECDH steps bring two advantages. It permanently binds the car part to the chassis' ECU by generating a mutually shared secret, unique to that association, as to prevent easy swapping of authentic parts. It also enables a very fast authentication (exposed later) skipping further certification verifications and using secret key algorithms which are much faster than the two-step certificate-based authentication. The DS28C40 stores the same shared secret (SECRET_S) in its memory with the use of “Compute and Write SHA-256 Secret” command, while the ECU stores the same secret in protected memory consequently completing the pairing as per Figure 6. If the part gets replaced by a new one, it is rejected by the ECU as the in-memory shared key used by the ECU to verify the part's response does not match the new part.

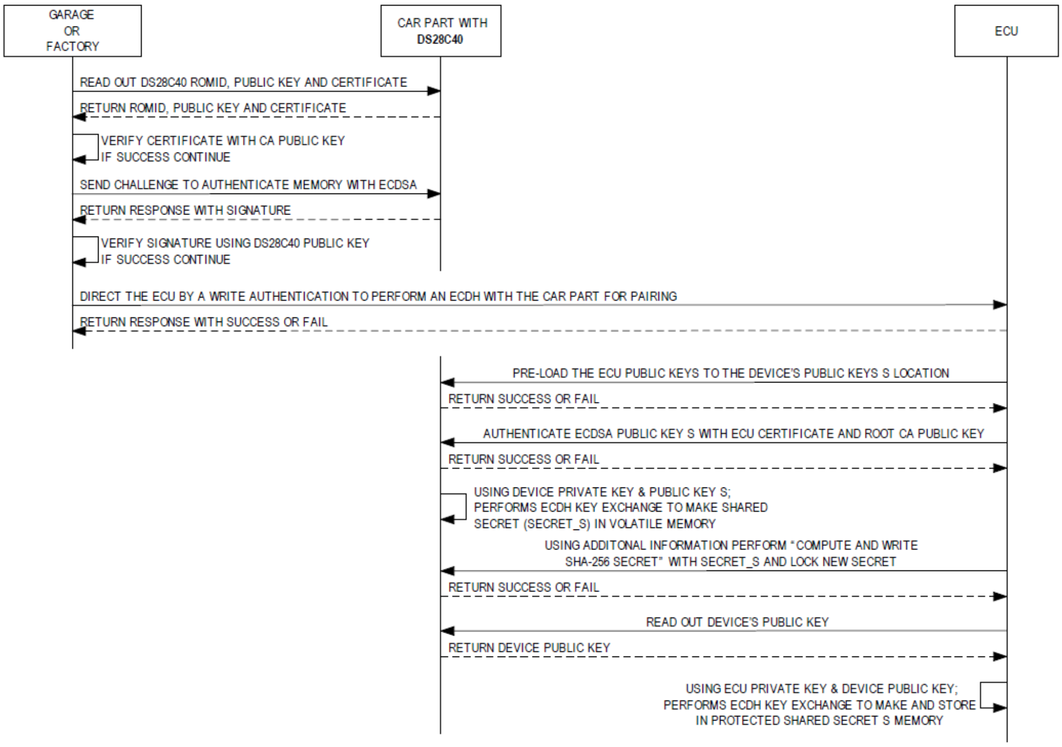

GARAGE
OR
FACTORY
CAR PART WITH
DS28C40
ECU
READ OUT DS28C40 ROMID, PUBLIC KEY AND CERTIFICATE
RETURN ROMID, PUBLIC KEY AND CERTIFICATE
VERIFY CERTIFICATE WITH CA PUBLIC KEY
IF SUCCESS CONTINUE
SEND CHALLENGE TO AUTHENTICATE MEMORY WITH ECDSA
RETURN RESPONSE WITH SIGNATURE
VERIFY SIGNATURE USING DS28C40 PUBLIC KEY
IF SUCCESS CONTINUE
DIRECT THE ECU BY A WRITE AUTHENTICATION TO PERFORM AN ECDH WITH THE CAR PART FOR PAIRING
RETURN RESPONSE WITH SUCCESS OR FAIL
PRE-LOAD THE ECU PUBLIC KEYS TO THE DEVICE'S PUBLIC KEYS S LOCATION
RETURN SUCCESS OR FAIL
AUTHENTICATE ECDSA PUBLIC KEY S WITH ECU CERTIFICATE AND ROOT CA PUBLIC KEY
RETURN SUCCESS OR FAIL
USING DEVICE PRIVATE KEY & PUBLIC KEY S;
PERFORMS ECDH KEY EXCHANGE TO MAKE SHARED
SECRET (SECRET_S) IN VOLATILE MEMORY
USING ADDITONAL INFORMATION PERFORM "COMPUTE AND WRITE
SHA-256 SECRET" WITH SECRET_S AND LOCK NEW SECRET
RETURN SUCCESS OR FAIL
READ OUT DEVICE'S PUBLIC KEY
RETURN DEVICE PUBLIC KEY
USING ECU PRIVATE KEY & DEVICE PUBLIC KEY;
PERFORMS ECDH KEY EXCHANGE TO MAKE AND STORE
IN PROTECTED SHARED SECRET S MEMORY

Figure 6. Pairing the DS28C40 by key establishment.

Various Part-to-Car Verification Schemes

Car parts and ECU previously paired using one of the explained methods in this document, such as certificate-based, static public key, shared secret key, or key establishment now provide a much stronger protection to the car's safety. Counterfeits, fraud, and attacks are thwarted by being able to mutually authenticate identity and life cycle state of the car part or the ECU. If any of the controlled assets cannot be trusted because it fails the verification process, then the ECU or the car part can cease to operate and keep the system in a safe state, for example, preventing the car to run, displaying alerts on the dashboard. Reverse authentication schemes allow car parts to cease functioning when they are not attached to a legit ECU (because of a man-in-the middle attack or an uncontrolled swapping of the part into another vehicle). Such scheme is also used to control write access to the car part's internal memory to preserve life cycle and configuration data from unwanted modification.

Identification and Authentication of Car Part by an ECU

An ECU must identify and authenticate a car part during normal usage to be assured it is genuine, and properly associated to the chassis. This can be done when engine starts, and periodically while the car is running.

Certificate-Based Authentication

In a certificate-based authentication, the ECU and the car part must share a common certification authority certificate. The document has explained the various certification delegation schemes, but eventually, the ECU must be able to verify the car part's certificate using the public key of the certification authority that issued the part's certificate. If the car part also needs to authenticate the ECU, then the car part must also be able to verify the ECU's certificate using the certification authority's certificate.

When using a DS28C40, the car part certificate can be stored in the DS28C40 memory (the ECU can retrieve it using a "Read Memory" command). The DS28C40 can store a certification authority certificate as per a specific format defined in the DS28C40 specifications. The ECU certificate can be verified by the DS28C40 using the "Authenticate ECDSA Public Key" command. Upon success, the ECU's public key is considered trusted. The certificate verification step (one way or both ways) is followed by the next step explained in Figure 7.

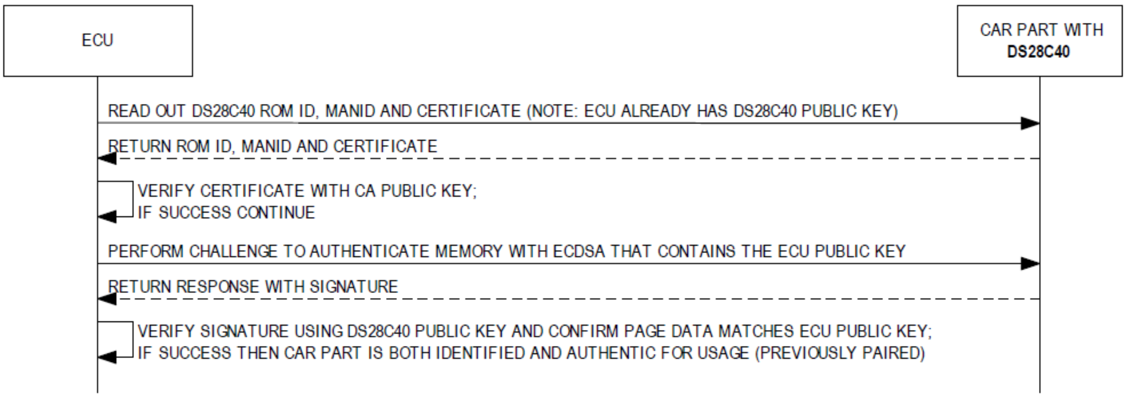

ECU
CAR PART WITH DS28C40
READ OUT DS28C40 ROM ID, MANID AND CERTIFICATE (NOTE: ECU ALREADY HAS DS28C40 PUBLIC KEY)
RETURN ROM ID, MANID AND CERTIFICATE
VERIFY CERTIFICATE WITH CA PUBLIC KEY; IF SUCCESS CONTINUE
PERFORM CHALLENGE TO AUTHENTICATE MEMORY WITH ECDSA THAT CONTAINS THE ECU PUBLIC KEY
RETURN RESPONSE WITH SIGNATURE
VERIFY SIGNATURE USING DS28C40 PUBLIC KEY AND CONFIRM PAGE DATA MATCHES ECU PUBLIC KEY; IF SUCCESS THEN CAR PART IS BOTH IDENTIFIED AND AUTHENTIC FOR USAGE (PREVIOUSLY PAIRED)

Figure 7. Verify public key and signature by ECDSA.

Static Public Key Authentication

To be considered authentic, devices need to prove the knowledge of the private key corresponding to the public key advertised in their certificates (counterfeiters never have access to genuine device's private keys). Possession of the right private key can be proven using the ECDSA algorithm in a challenge-response portion of the protocol as per Figure 7. To that end, the ECU exercises the Compute and Read Page Authentication sequence of the DS28C40. The ECU sends a challenge to the DS28C40 through this command.

The DS28C40 digitally signs the random number appended to additional information with an ECDSA signature operation using its private key. The resulting signature is returned to the ECU which performs an ECDSA verification operation over the same data and using the car part's public key. Trust in the public key used for verification may have been established in a prior certificate verification step as exposed above, or through a trusted installation of the part's public key into the ECU's memory in a static manner. Upon success, the ECU gets a formal proof that both ends used the same data and the car part's private key matches the public key used by the ECU. The public key cryptography has the advantage of not sharing secret information between the various devices. However, trust in public keys must be guaranteed, either through a certificate based scheme or by using a trusted public key exchange process as exposed in the first part.

Shared Secret-Based Authentication

Whether the shared secret is directly installed or established through ECDH, this car part authentication process relies on the execution of the challenge-response protocol. The DS28C40 uses the HMAC-SHA256 as a message authentication code (MAC). In this protocol, the ECU sends a random challenge to the car part as a parameter of the "Compute and Read Page Authentication" command provided by the DS28C40 sitting on the car part. The DS28C40 computes the HMAC-SHA256 of the challenge (and other data appended) using the shared secret and returns the resulting MAC. This process is shown in Figure 8.

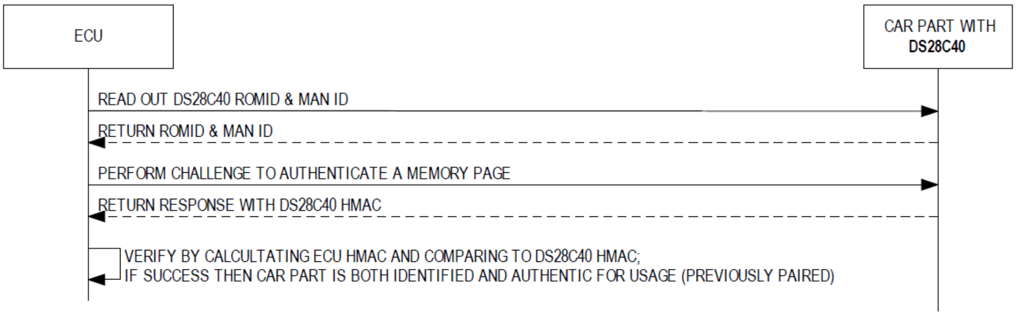

ECU
CAR PART WITH DS28C40
READ OUT DS28C40 ROMID & MAN ID
RETURN ROMID & MAN ID
PERFORM CHALLENGE TO AUTHENTICATE A MEMORY PAGE
RETURN RESPONSE WITH DS28C40 HMAC
VERIFY BY CALCULTATING ECU HMAC AND COMPARING TO DS28C40 HMAC;
IF SUCCESS THEN CAR PART IS BOTH IDENTIFIED AND AUTHENTIC FOR USAGE (PREVIOUSLY PAIRED)

Figure 8. Verify HMAC.

The ECU performs the exact same calculation over the same data with its own version of the shared key and generates a second MAC. If both MAC values match, it confirms that both the data and shared secret keys match on both ends, proving the accessory is valid. Now, usage of the car part can be authorized. The advantage of these two methods is that the computation is 12x quicker than the public key method that uses ECDSA.

Authentication (Attestation) of Car Part's Memory Content by an ECU

Attestation of a car part's memory content complements the identification/authentication and allows to get tamper-proof evidence of its various properties, such as life cycle information, calibration and settings, maintenance, and manufacturing steps or other arbitrary OEM-defined information. This attestation is, in fact, executed simultaneously with the exposed identification and authentication method mentioned above where the part's memory content is signed (through ECDSA or HMAC-SHA256) together with the incoming ECU challenge while executing the "Compute and Read Memory Authentication" command, therefore, proving its origin and authenticity. If a single bit of the DS28C40's memory data is modified in transit between the car part and the ECU, or if the signing key is invalid, the ECU notices as it fails to verify the signature of the data, thus blocking all tampering of forgery attempts. The data is usually retrieved from the car part's memory by issuing "Read Memory" commands to the DS28C40. It can also be simply "assumed" by the ECU, eliminating the need to read the data every time an authentication must be performed.

On a DS28C40 IC, the memory is split into several fixed-length pages. The same authentication process can be repeated with any of the pages. Also, if needed, encryption can be added with more steps to prevent eavesdropping of the car part's memory content while in transit.

Secure Writing into Car Part's Memory Pages

The car part's memory content is obviously an asset to secure; therefore, a strict write access control is needed to modify data, such as calibration, settings, and/or life cycle information. To achieve this, the former pairing and authentication schemes are leveraged. This section discusses a few ways in which the car part's memory can be protected.

Writing of Calibration/Life Cycle Data

During the servicing of a vehicle's subsystem at the garage or factory, memory write operations can be accomplished and controlled by an accredited actor. The DS28C40 offers a memory write access control using the ECDSA algorithm or using the shared-secret HMAC-SHA256 algorithm. Encrypted write operation is possible, although it is not addressed in this application note. First, a read out of the current target memory page content is performed as per Figure 9. Then:

For a Public Key-Based Authentication:

- The garage/factory runs an "Authenticate ECDSA Public Key" command to authenticate their own certificate. Provided the DS28C40 was initialized with a CA certificate that matches the garage's certificate, the DS28C40 trusts the proposed certificate and use the corresponding public key for the subsequent ECDSA verification operation.
- The garage/factory runs an "Authenticate ECDSA Write Memory" command, digitally signing both the current and new memory page data, using their private key. Various options can be implemented at the OEM's discretion in this case: The garage/factory can possess the private key locally in a "secure box," or the authentication process can be run online with an OEM server, keeping private keys remotely stored in secure infrastructures.
- The DS28C40 verifies the signature coming within the "Authenticated ECDSA Write Memory" command using the previously verified public key received in the "Authenticate ECDSA Public Key" command. Upon success, the memory content is updated, meaning that both the keys and the information included in the digital signature calculation match, avoiding override of access control and manipulation of the data in transit.

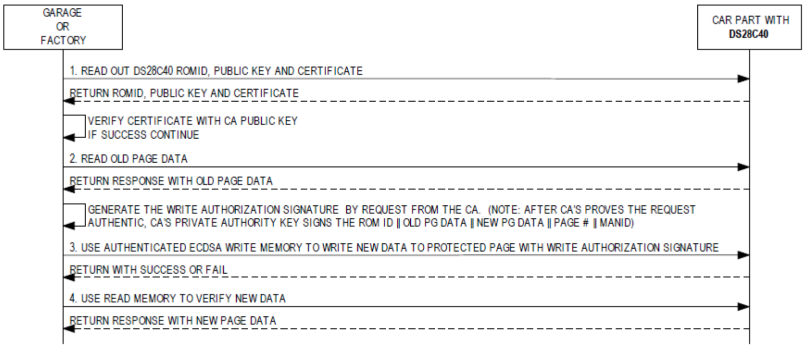

GARAGE OR FACTORY
CAR PART WITH DS28C40
1. READ OUT DS28C40 ROMID, PUBLIC KEY AND CERTIFICATE
RETURN ROMID, PUBLIC KEY AND CERTIFICATE
VERIFY CERTIFICATE WITH CA PUBLIC KEY IF SUCCESS CONTINUE
2. READ OLD PAGE DATA
RETURN RESPONSE WITH OLD PAGE DATA
GENERATE THE WRITE AUTHORIZATION SIGNATURE BY REQUEST FROM THE CA. (NOTE: AFTER CA'S PROVES THE REQUEST AUTHENTIC, CA'S PRIVATE AUTHORITY KEY SIGNS THE ROM ID ‖ OLD PG DATA ‖ NEW PG DATA ‖ PAGE # ‖ MANID)
3. USE AUTHENTICATED ECDSA WRITE MEMORY TO WRITE NEW DATA TO PROTECTED PAGE WITH WRITE AUTHORIZATION SIGNATURE
RETURN WITH SUCCESS OR FAIL
4. USE READ MEMORY TO VERIFY NEW DATA
RETURN RESPONSE WITH NEW PAGE DATA

Figure 9. Writing data to DS28C40 setup with a public key.

For a Shared-Secret Key-Based Authentication:

- The garage/factory runs an "Authenticate SHA256-Write" command, calculating an HMAC-SHA256 over both the current and new memory page data, using their shared secret key. Various options can be implemented at the OEM's discretion in this case: The garage/factory can possess the shared key locally in a "secure box," or the authentication process can be run online with an OEM server, keeping secret keys remotely stored in secure infrastructures.
- The DS28C40 verifies the MAC coming within the "Authenticated ECDSA Write" command using the shared secret. Upon success, the memory content is updated, meaning that both the shared secret keys and information included in the MAC calculation match, avoiding override of access control and manipulation of the data in transit. The process is shown in Figure 10.

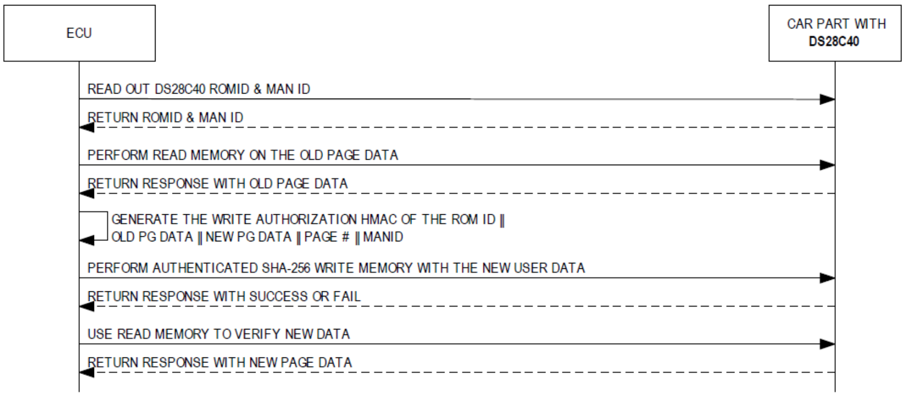

ECU
CAR PART WITH DS28C40
READ OUT DS28C40 ROMID & MAN ID
RETURN ROMID & MAN ID
PERFORM READ MEMORY ON THE OLD PAGE DATA
RETURN RESPONSE WITH OLD PAGE DATA
GENERATE THE WRITE AUTHORIZATION HMAC OF THE ROM ID ‖ OLD PG DATA ‖ NEW PG DATA ‖ PAGE # ‖ MANID
PERFORM AUTHENTICATED SHA-256 WRITE MEMORY WITH THE NEW USER DATA
RETURN RESPONSE WITH SUCCESS OR FAIL
USE READ MEMORY TO VERIFY NEW DATA
RETURN RESPONSE WITH NEW PAGE DATA

Figure 10. Writing data to DS28C40 setup with HMAC keys.

Cryptographic Activation of Car Part by an ECU

In this scheme, the car part authenticates the ECU. Using the DS28C40, this translates into performing an authenticated write operation described above in the document to control an output pin of the DS28C40. Without the proper credentials, controlling the output pin is impossible. The output pin can control the level of a signal which may be partially or completely disabling the car part operation, whatever option is appropriate from a safety perspective.
When using a DS28C40 to control the activation of a car part, an ECU must prove that it owns a legitimate shared secret or a private key (both schemes are possible). See Secure Writing into Car Part's Memory Pages section as it works the same way but writing to a special virtual memory page that controls the GPIO state as per Figure 11.

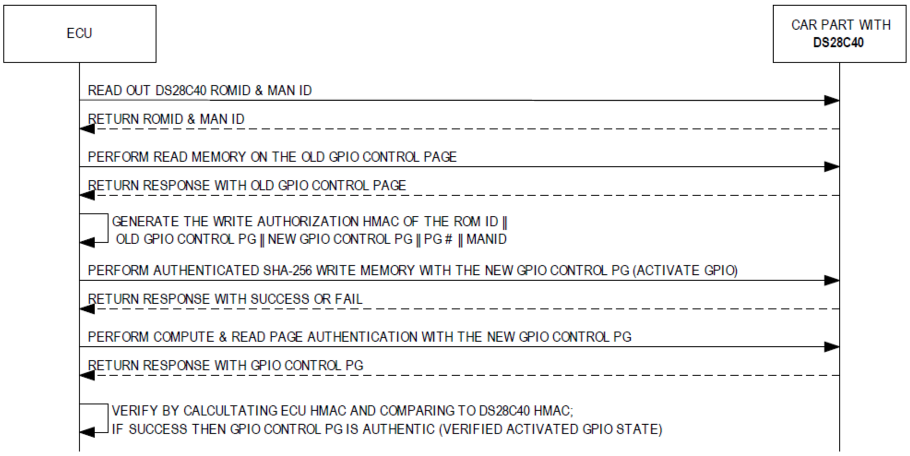

ECU
CAR PART WITH DS28C40
READ OUT DS28C40 ROMID & MAN ID
RETURN ROMID & MAN ID
PERFORM READ MEMORY ON THE OLD GPIO CONTROL PAGE
RETURN RESPONSE WITH OLD GPIO CONTROL PAGE
GENERATE THE WRITE AUTHORIZATION HMAC OF THE ROM ID ‖ OLD GPIO CONTROL PG ‖ NEW GPIO CONTROL PG ‖ PG # ‖ MANID
PERFORM AUTHENTICATED SHA-256 WRITE MEMORY WITH THE NEW GPIO CONTROL PG (ACTIVATE GPIO)
RETURN RESPONSE WITH SUCCESS OR FAIL
PERFORM COMPUTE & READ PAGE AUTHENTICATION WITH THE NEW GPIO CONTROL PG
RETURN RESPONSE WITH GPIO CONTROL PG
VERIFY BY CALCULTATING ECU HMAC AND COMPARING TO DS28C40 HMAC;
IF SUCCESS THEN GPIO CONTROL PG IS AUTHENTIC (VERIFIED ACTIVATED GPIO STATE)

Figure 11. Activation by write authentication to GPIO control.

Summary

Guaranteed car part performance and safe operation can be better achieved by implementation of the various pairing options discussed. These options can be best achieved by devices such as DS28C40 along with the usage/knowledge of the cryptography schemes presented for ECDSA and HMAC-SHA. A simplified summary of benefits is shown in Table 4.

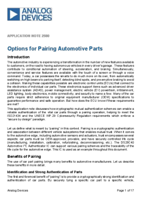
Table 4. Benefits of Different Options

SCHEME	KEY INSTALL PROCESS	PAIRING PROCESS	ID AND STRONG AUTHENTICATION OF PARTS	STRONG AUTHENTICATION OF LIFE CYCLE DATA	SECURE LIFE CYCLE DATA WRITE	CRYPTO-GRAPHIC ACTIVATION OF CAR PARTS	COMPUTE LATENCY
Certificate-Based	Involved (CA is required)	Easy	Strong	Strong	Strong	Strong	Slow
Static Public Keys	Easy (Install facility must be trusted.)	Easy (A drawback is pairing facility must be trusted.)	Medium (If a rogue player can control public keys, then a new part can be associated.)	Medium	Medium	Medium	Medium (2x faster as Certificate- based; no certificate verification)
Shared Secret Key Directly	Easy (Install facility must be trusted.)	Easy (A drawback is pairing facility must be trusted.)	Strong	Strong	Strong	Strong	Fast (12x faster than Certificate- based)
Key Establishment	Involved (CA is required)	Involved (ECDH required)	Strong	Strong	Strong	Strong	Fast (Same as Shared Secret Key Directly)

References/Other Resources

Haight, Michael. Solution Guide 7632, DeepCover Secure Automotive Authenticator Solution Guide.
Design Solutions No. 56, Trust Your Digital Certificates—Even When Offline.
For additional details, refer to the DS28C40/DS28E40/DS2478 Data Sheets, DS28C40/DS28E40/DS2478 Security User's Guides, and the DS28C40 EV kit/DS28E40 EV kit Data Sheets.

Documents / Resources

	**ANALOG DEVICES DS28C40 DeepCover Automotive I2C Authenticator** [pdf] User Manual DS28C40 DeepCover Automotive I2C Authenticator, DeepCover Automotive I2C Authenticator, Automotive I2C Authenticator, Authenticator

ANALOG DEVICES

APPLICATION NOTE 2580

Options for Pairing Automotive Parts

Introduction

The automotive industry is experiencing a transformation in the number of new features available to customers, on the road to having autonomous vehicles in every driver's garage. These features contribute to conditional automation of steering, acceleration, and braking. Simultaneously, convenience and service features are available with the touch of a screen or through a voice command. Today, a car possesses the smarts to do much more on its own, from automatically switching on high beams to parking itself, detecting blind spots, and pre-emptive braking to avoid a collision. Making these capabilities possible are electronic control units (ECUs) that connect to the electronics of individual car parts. These electronics support items such as advanced driver assistance system (ADAS), power management, electric vehicle (EV) powertrain, infotainment, LED lighting, body electronics, mobile connectivity, and security to name a few. Many of the car parts require strict adherence to original equipment manufacturer (OEM) specifications to guarantee performance and safe operation. But how does the ECU know if these requirements are met?

This application note discusses how cryptographic mutual authentication schemes can enable a reliable authentication of the vital car parts through a pairing operation, helping to meet the ISO21434 and the UNECE WP.29 Cybersecurity Regulation requirements which enforce a "secure by design" paradigm.

Let us define what is meant by "pairing" in this context. Pairing is a cryptographic authentication and association between different vehicle subsystems that enables mutual trust. When it comes to the automotive edge, including automotive sensors and actuators, trust encompasses several aspects: car parts must be OEM-approved, provable, and have securely controlled life cycle (manufacturing, installation, calibration, refurbishing, decommissioning, etc.). The DS28C40 Automotive I²C Authenticator IC can support various pairing schemes and the traceability of the life cycle for the automotive edge. This IC is used as an example throughout this document.

Benefits of Pairing

The use of car part pairing brings many benefits to automotive manufacturers. Let us describe these benefits in more detail.

Identification and Strong Authentication of Parts

The first and foremost benefit of "pairing" is to provide a cryptographically strong identification and authentication of car parts. By securely tying a specific car part to a specific vehicle,

Analog Devices Page 1 of 17

References

- User Manual