Application Note

FAQ: On the use of S0, S2 and Supervision CC in product design and deployments

Document No.:	APL13434
Version:	3
Description:	Includes topics on protection levels, firmware update, performance considerations, etc.
Written By:	ABR;JRM;NTJ;BBR
Date:	2018-03-05
Reviewed By:	BBR
Restrictions:	Approved Partners Only

Approved by:

Date	CET	Initials	Name	Justification
2018-03-05	11:52:49	NTJ	Niels Thybo Johansen	

REVISION RECORD

Doc. Rev	Date	By	Pages affected	Brief description of changes
1B	20151110	ABR	ALL	First revision
1C	20151116	ABR	most	Added NTJ comments and proposals.
3	20180302	BBR	All	Added Silicon Labs template

Table of Contents

1 WHAT IS A MAN-IN-THE-MIDDLE ATTACK?

In a man-in-the-middle attack, the network controller and a node think that they are communicating directly, but in reality the network controller talks to the attacker and the attacker talks to the node.

The attack may require advanced jamming and capture of commands, but if the attacker manages to set up the system, there is no way to detect the attack via normal commands. The node responds correctly to commands from the network controller and everything works.

The only way to detect a man-in-the-middle attack is to confirm vital key information via an out-of-band (OOB) interface, e.g. a QR code or PIN code. The attacker has no way of guessing this information as he has no access to the (OOB) interface.

2 WHAT IS A DELAY ATTACK?

A delay attack is a special variant of the man-in-the-middle attack (see 1).
A classic use of the delay attack is with car keys. Imagine the car has just been parked.

The car owner pushes the button but an attacker jams the receiver of the car while he records the (first) message sent by the key.

Nothing happens, so the car owner pushes the button one more time.
The attacker still jams the receiver of the car while he records the second message sent by the key.
Now, the attacker sends the first message to the car, which accepts the rolling code of the message.
The car locks the doors and the car owner leaves.

The attacker can now send the second message to the car to make the car unlock again.

Mechanisms such as two-way communication using authenticated positive acknowledgements with sequence numbers effectively prevent delay attacks.

3 WHAT IS THE TEMPORARY SECURE CONNECTION USED FOR?

The temporary secure connection is used to transfer one or more network keys (S0, S2 Access Control Class, S2 Authenticated Class, S2 Unauthenticated Class). The temporary secure connection is established Elliptic Curve Diffie-Hellman key exchange between the network controller and the joining node.

If the network controller does not know the identity of the joining node to the network, it may unintentionally add an attacking node to the network instead of the true joining node.
Handing out the network key to an attacker compromises the security of door locks and alarm systems.
Therefore, the S2 Authenticated Class and S2 Access Control Class require QR code or PIN code authentication before the network keys are transferred to the joining node.

4 WHAT IS DIFFIE-HELLMAN KEY EXCHANGE USED FOR?

While Key Exchange and cryptography generally relies on sophisticated mathematics, the actual principle of Diffie-Hellman Key Exchange is straightforward: It is relatively simple to multiply very large prime numbers but it is very difficult reverse the calculation if one does not know one of the factors.

The direct parallel to daily life is that it is easy to mix two colors of paint, but it is rather hard to separate the two colors again.

In that metaphor, adding Elliptic Curve mathematics to the equation corresponds to creating a relatively small palette which still can hold an incredibly large number of colors. Thus, Elliptic Curves allow for smaller keys while maintaining the same high protection level.

Recommended videos: https://www.khanacademy.org/computing/computer-science/cryptography/modern-crypt/v/diffie-hellman-key-exchange-part-1diffie-hellman-key-exchange-part-1, diffie-hellman-key-exchange-part-2.

5 WHY DO I NEED A QR CODE OR PIN ENTRY?

Scanning a QR code or entering a PIN code both represents the transfer of information out of reach to an attacker who listens to the wireless communication. In communication security, this is called out-of-band (OOB) authentication. The QR code represents a Device Specific Key (DSK) in S2 while the PIN code is an extract of the complete DSK.

The Device Specific Key allows the network controller to actively authenticate the joining node by comparing the Device Specific Key to the truncated Diffie-Hellman public key that the joining node advertises over the wireless connection.
The authentication allows the network controller to reject handing out the network key to rogue nodes.

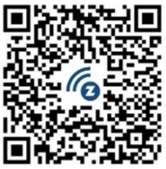
If the device requests membership of the Access Control Class, it sets the first two bytes of the truncated public key to zero to force the user to enter the correct byte values; either from scanning the QR code or manually entering the first 5 decimal digits.

6 WHAT IS THE DSK?

The DSK is a part of the public key. The DSK is printed physically on the device – or it can be shown on a display if that is available.

The DSK is a truncated version of the public key. The public key is 32 bytes long.
The DSK is the first 16 bytes of the public key.
The PIN code is the first 2 bytes of the public key.

Two representations are used for the DSK: ASCII Text and QR code. When scanned, it can be seen that the QR code contains exactly the same characters as the text line.



zws2dsk:34028-23669-20938-46346-33746-07431-56821-14553

The first digit block is underlined to highlight that this is the PIN code.

The DSK information must be made available on devices intended for the S2 Access Control Class and the S2 Authenticated Class. Low-cost devices with limited security risk intended for the S2 Unauthenticated Class do not have to make the DSK information available.

7 CAN THE DSK BE VERIFIED?

A non-included S2 device broadcasts its public key when powered. This allows test personnel to verify that the QR code printed on a device actually matches the public stored electronically inside the device.

Access Control devices set the first two bytes of the transmitted public key to zero. Thus, the full DSK cannot always be determined from the transmitted public key. Thus, the production line should maintain a database of programmed public keys so that the DSK may be reconstructed by using the 14 non-modified public key bytes to look up the complete 16 bytes public key in the database.

8 WHAT IS THE S2 UNAUTHENTICATED CLASS?

The Z-Wave Security 2 Command Class supports many application spaces. The S2 Unauthenticated class enables secure applications at the low end of security scale provided by S2.
While the S2 Authenticated class is less secure than other S2 classes, it still represents a significant improvement over the protection level that can be achieved with the original Z-Wave Security Command Class (S0).

The S2 Unauthenticated class enables the deployment of simple networks with very constrained network controllers. One example is a wood cabin, where a battery powered wireless wall switch controls a few LED bulbs running off a car battery and a solar panel. The wall switch also acts as the network controller, but as it has no QR scanner and no keypad for decimal entry, it is more convenient to only assign the S2 Unauthenticated class key to the LED bulbs.

The S2 Unauthenticated class also allows for low-cost variants of devices such as LED bulbs and dimmers as it eliminates the production logistics involved in tracking the individual product during production and attaching QR code and PIN code on the outside of the finished product.

Z-Wave certification will prohibit some products from operating via the S2 Unauthenticated class. This includes gateways and door locks. In other cases, manufacturers may decide that their particular application needs a protection level of at least the S2 Authenticated class.

Products may be designed to accept multiple S2 classes. For instance, a full-functional LED bulb may accept joining the S2 Unauthenticated class as well as the S2 Authenticated class.

9 DOES ANY Z-WAVE PLATFORM SUPPORT SECURITY 2?

Code space requirements for Security 2 enabled development kits dictate that that a Z-Wave 500 series chip is used. 1Mbits of Serial Flash memory is needed for support of Over-the-Air (OTA) firmware update.

10 DO SECURITY 2 KEYS HAVE TO BE STORED DURING PROGRAMMING?

As ECDH public and private keys are not a part of the firmware image, the keys may be stored at the same time as firmware is programmed into the device or at a later time during testing if that is more convenient in the actual production flow.

11 CAN I TRUST THAT A NON-SECURE ACKNOWLEDGED COMMAND IS EXECUTED?

The Z-Wave Acknowledgement message (Ack) indicates that a non-secure Z-Wave frame was delivered.

Most likely, the Ack is sent by the receiving node but theoretically, an attacker may be jamming the intended destination and returning the Ack on behalf of the intended destination.

Thus, the reception of an Ack does not guarantee that a transmitted command was actually received by the intended recipient.

Further, the Ack does not advertise any application status. Mechanical issues or other device limitations may prevent a device from executing a received command.

Transport protocols like the Firmware Update Command Class employs their own application status synchronization mechanisms and are therefore resistant to above phenomena.

Control applications will have to request the application status subsequently.

12 CAN I TRUST THAT AN S0 SECURE ACKNOWLEDGED COMMAND IS EXECUTED?

The S0 command sequence involves a NonceGet-NonceReport handshake before the encrypted command is transmitted. This guarantees that if the command reaches the intended destination, then the command can be decrypted.

The handshake also ensures that a command cannot be stored and re-used at a later time (also known as a replay attack).

As per 1, a sender cannot trust that an acknowledged S0 command is executed but it can prevent attackers from injecting commands.

Control applications will have to request the application status subsequently – also via S0 secure communication. Only a secure positive response from the application guarantees that the command was actually executed.

13 CAN I TRUST THAT AN S2 SECURE ACKNOWLEDGED COMMAND IS EXECUTED?

The S2 command sequence uses a rolling code mechanism to obtain low latency transfer of secure messages. This provides low latency for transport protocols but it does not guarantee that the intended destination can decrypt the command.
S2 employs a fall-back option of using a NonceGet-NonceReport handshake to re-establish a shared rolling code between sender and receiver. This guarantees that if the command reaches the intended destination, then the command can be decrypted.

Both the rolling code and handshake both ensures that a command cannot be stored and re-used at a later time (also known as a replay attack).

As per 1, a sender cannot trust that an acknowledged S2 command is executed but it can prevent attackers from injecting commands.

Control applications could request the application status subsequently but S2 offers integration with the Supervision Command Class. Popularly speaking, the Supervision Command Class allows the initial control command and the decryption confirmation to be collapsed into one command.
The Supervision Command Class further allows a receiving node to advertise if it is prevented from executing a received command or if the command was executed successfully.

14 HOW SECURE IS A SECURE DEVICE?

The Security 2 Command Class uses state-of-the-art technologies for key exchange and encryption. This does however not protect a device from being incapacitated in more simple ways.
A movement sensor may be removed, covered or rotated it so that it looks into a wall. A key pad may have a small keylogger added to it.

A light bulb may not need additional protection but alarm sensors and door locks should employ mechanisms to detect mechanical movement, detect when the device is removed from its mounting bracket and when the battery enclosure is opened.

15 DOES S2 NEED THE SUPERVISION COMMAND CLASS?

The S2 key exchange and encryption algorithms do not depend on the Supervision Command Class. There are however several security related reasons for using the Supervision Command Class together with S2.

The Supervision Command Class provides a positive confirmation to a sending node that the intended destination could decrypt the command.

Thanks to the positive confirmation, a sending node does not have to wait for the rare case of an error indication because the rolling code is out of sync between the sender and the receiver. Not only does this allow a sending node to continue sending other commands; thus improving performance. It also prevents an attacker from mounting a so-called delay attack (refer to 1).

As a bonus, the Supervision Command Class can convey updated application status when available. This saves network bandwidth and improves UI responsiveness compared to solutions which use polling to track application status.

16 CAN A NON-SECURE NODE BE OVER-THE-AIR UPGRADED TO S2?

Provided that the manufacturer offers a firmware image, the node can be updated – but it will have to be excluded from the network and re-included.

The re-inclusion step is needed to establish a trust relationship between the network controller and individual node before the network key is transferred.
Further, the lack of a Device Specific Key (QR code and PIN code) forces the upgraded node to operate only in the S2 Unauthenticated class.

17 CAN AN S0 NODE BE OVER-THE-AIR UPGRADED TO S2?

There may exist constrained platforms which do not support Over-the-Air update, or which cannot be updated to S2 because of missing available code space. Refer to the question “9 Does any Z-Wave platform support Security 2?”

Provided that the manufacturer offers a firmware image, the node can be updated – but it will have to be excluded from the network and re-included.

While the AES-128 encryption used by the Security Command Class (S0), the S0 key exchange is vulnerable during a short time window when a node joins the network. The S2 temporary secure connection does not suffer from the same vulnerability and is therefore much more secure. This requires a re-inclusion.

The re-inclusion step is needed to establish a trust relationship between the network controller and individual node before the network key is transferred.
Further, the lack of a Device Specific Key (QR code and PIN code) forces the upgraded node to operate only in the S2 Unauthenticated class.

18 CAN SECURITY CLASSES BE MIXED?

A joining device may be granted keys for several security classes, e.g. S2 Authenticated Class and S2 Unauthenticated Class. However, the device only accepts commands received with the highest class key (in this example: the S2 Authenticated Class).

There is one exception from this rule: Gateways. A gateway may employ a “smart rules” module which maps incoming commands from certain node to outgoing commands to other nodes. Ultimately, such a module is able to map non-secure Basic Set commands to S2 Access Control Class protected Door Lock (Open) commands. Such a mapping effectively downgrades the protection level of the door lock to non-secure. Therefore, the user should always be warned when trying to create rules which downgrades the protection level

The Security 2 framework allows that the joining device sends commands to other nodes via different security classes. The Beta release of Security 2 does not include association commands which allow such multi-class association configurations.

19 CAN SECURITY CLASS REQUESTS BE OVERRULED?

Yes, security class requests can be overruled; but only to a smaller subset.
A controller cannot grant access to an S2 Security Class that the joining does not request.

If a door lock requests the S2 Access Control Class only, a controller cannot force the door lock to operate in the S2 Authenticated Class.

If a low-cost light bulb is designed to request the S2 Unauthenticated Class only, a controller cannot force the light bulb to operate in the S2 Authenticated Class (which it would not be able to anyway if it does not have a DSK printed on it).

On the other hand, a full-featured light bulb may request both the S2 Unauthenticated Class and the S2 Authenticated Class. In this case, a controller may grant operation in the S2 Authenticated Class to allow the light bulb to interoperate directly with sensors and wall switches.

Z-Wave certification will prohibit some products from operating via the S2 Unauthenticated class. This includes gateways, sensors and door locks. In other cases, manufacturers may decide that their particular application needs a protection level of at least the S2 Authenticated class.

20 CAN I BLACKLIST A NODE IF IT IS LOST OR STOLEN?

If a key fob is lost outside the house, the user may add the NodeID of lost device to the list of blacklisted devices in the gateway. If the device is ever activated, e.g. by a stranger in the street, the gateway ignores commands from the key fob. The gateway may log the unauthorized usage attempt and optionally send a mail or a text message to alert the user.

As the lost device is not removed from the network as a failing device, there is no risk that the actual NodeID is re-assigned to another new node.

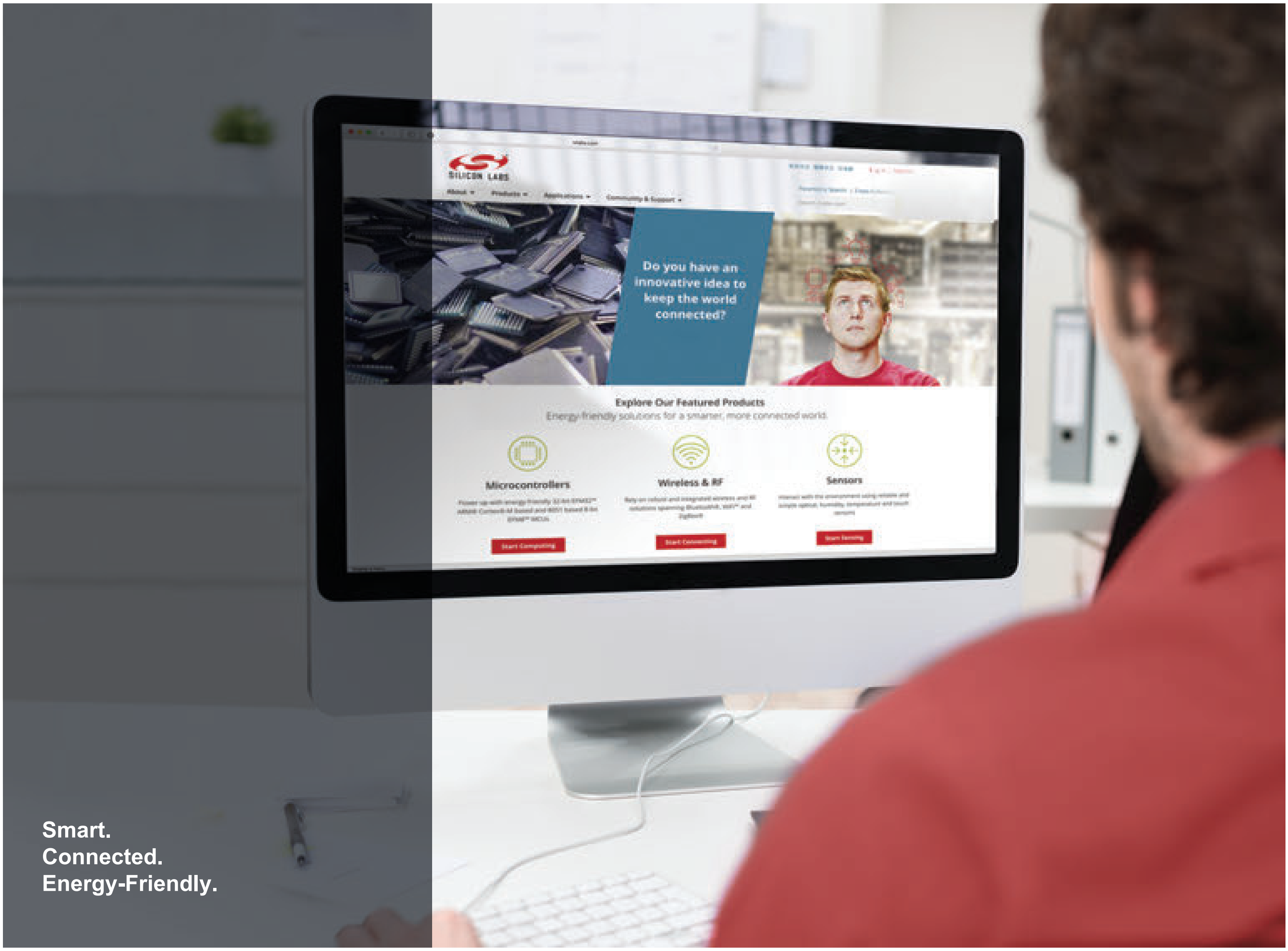
The above text makes a few assumptions:

- The key fob sends commands to the gateway; not directly to the door lock
- The gateway maps key fob commands to the door lock
- The gateway implements a blacklist where lost devices can be added – and a UI to that blacklist
- The gateway allows the user to assign a name to joining devices so that the user can identify the key fob – and move it to the blacklist

REFERENCES

[1] SDS12657 Z-Wave Command Class Specification, A-M

[2] SDS12652 Z-Wave Command Class Specification, N-Z


SILICON LABS
Do you have an innovative idea to keep the world connected?
Explore Our Featured Products
Energy-friendly solutions for a smarter, more connected world.
Microcontrollers
Wireless & RF
Sensors
Start Computing
Start Connecting
Start Sensing
Smart.
Connected.
Energy-Friendly.

Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

Disclaimer
Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Labs shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

Trademark Information
Silicon Laboratories Inc.® , Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, ISOmodem®, Micrium, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, Z-Wave and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com