Protecting IoT data with Tokenized Keys

The fast growth of Internet of Things (IoT) is well documented. Typical to IoT is large number of sensors and actuators connected wirelessly. These devices generate and consume wirelessly transferred data that is exposed and vulnerable. In order to secure that data, cryptography is used to perform encryption, decryption and authentication. Standard hardware and software employ algorithms such as AES-128 with encryption keys to provide IoT data protection.

In typical scenario, packets of data are encrypted by an originator with a shared key, transmitted over the air as encrypted packets and then decrypted by a receiver with the same shared key. The shared key, for most implementation, is static and does not change frequently enough. This is a known weak point!

Wireless standards, such as Bluetooth© Mesh, Zigbee©, Z-Wave© and others, protect the static keys by adding counters to each packet. Bluetooth© Mesh 5.0 adds an option for storing up to 4K shared keys so that they can be rotated over time.

Enter **"Tokenized Encryption Key (TEK)"**: Now there is a new way of availing a large number of encryption keys with small storage and minimal processing overhead. Having a large number of encryption keys, rather than few keys or single key, makes it that much harder to guess the correct encryption key and attack the protected data.

As an implementation example, quadrillion (1e+15) keys are stored in 1KBytes of memory. This amount of storage is compatible with today's IoT devices and their limited resources. Managing TEKs, benefits and implications are the subject of this writing.

Best practice of securing wireless sensors

Figure 1 depicts best practice (pre TEK) of wireless data protection using Hardware Security Module - HSM:

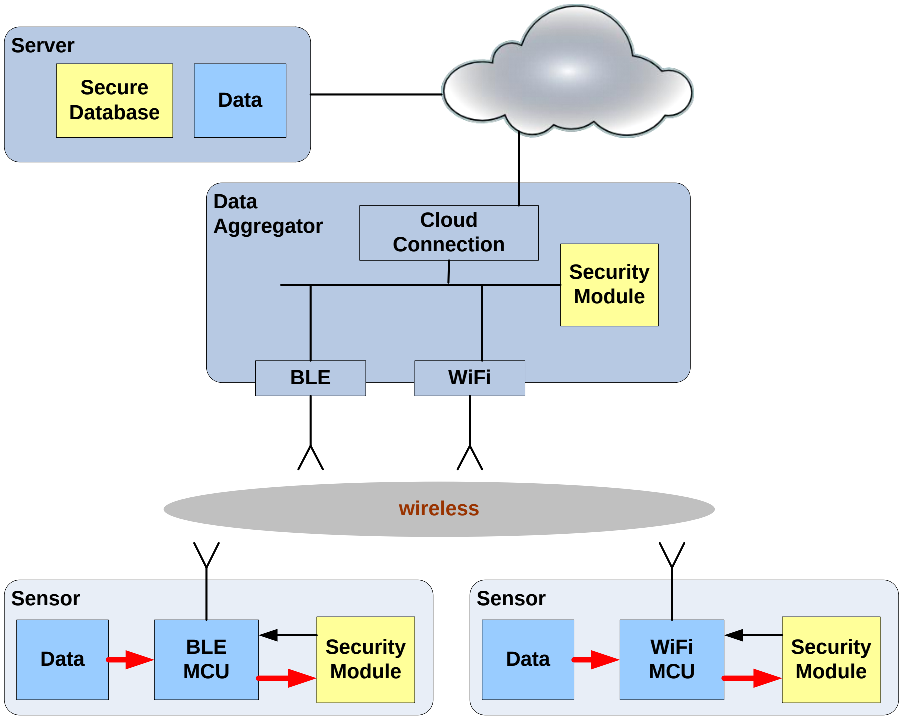
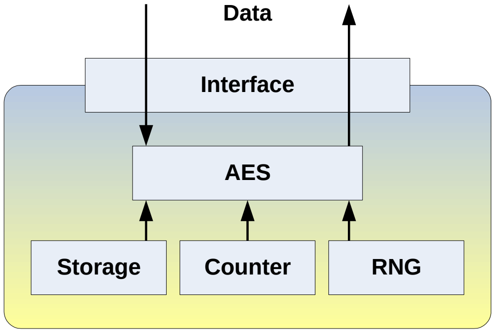


Figure 1: data protection using security module

Unprotected data, collected from sensors is encrypted inside the HSM using encryption keys stored in the module. The encryption keys are never exposed as all the action is done inside the module. Encrypted data is then transferred by the data aggregator to a server. The server uses its database to decrypt the data and present it to a user.

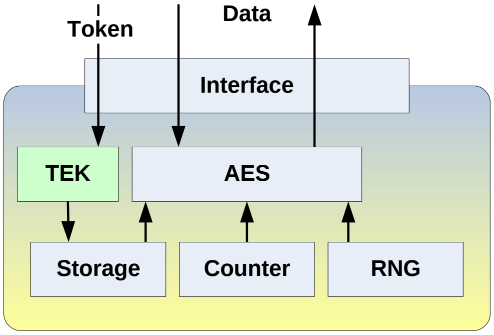
Hardware Security Module (HSM) description



Hardware Security Module (HSM) is a hardware device designed to minimize vulnerability of sensitive data. Typical module stores private keys, certificates and other private data. Advanced Encryption Standard (AES) engine is included to encrypt, decrypt and authenticate. Random Number Generator (RNG) and Counter are included to assemble wireless packets. See Figure 2. Stored private data is not accessed directly. Stored keys are fed directly to the AES engine and are never exposed outside the module. Private data is written once to the module One Time Programmable (OTP) memory during manufacturing.

Figure 2: Security Hardware Module

Adding TEK to Hardware Security Module



TEK expands the number of available encryption keys stored within HSM by adding a logic block depicted in Figure 3 as TEK. Tokens are passed to the TEK block via the HSM Interface block for processing. TEK presents to the AES engine new keys in a way similar to the method used in Figure 2 (HSM without TEK).

Figure 3: Adding TEK to Security HW Module

TEK authentication solution example

Selection of a specific key is handled by a dynamic token. The value of this token can be different for each packet of data transferred at IoT rates. Current IoT devices, with TEK hardware added, can support this feature for Bluetooth Smart© (Low Energy), Zigbee©, Z-Wave©, LoRa and others. The following generic wireless message exchange example depicted in Figure 4, shows an Access point authenticating a Sensor Device. A single message exchange is all that is needed. There is no key exchange and the over-the-air data is encrypted. If mutual authentication is desired, then additional exchange with the Sensor Device initiating a message is added.

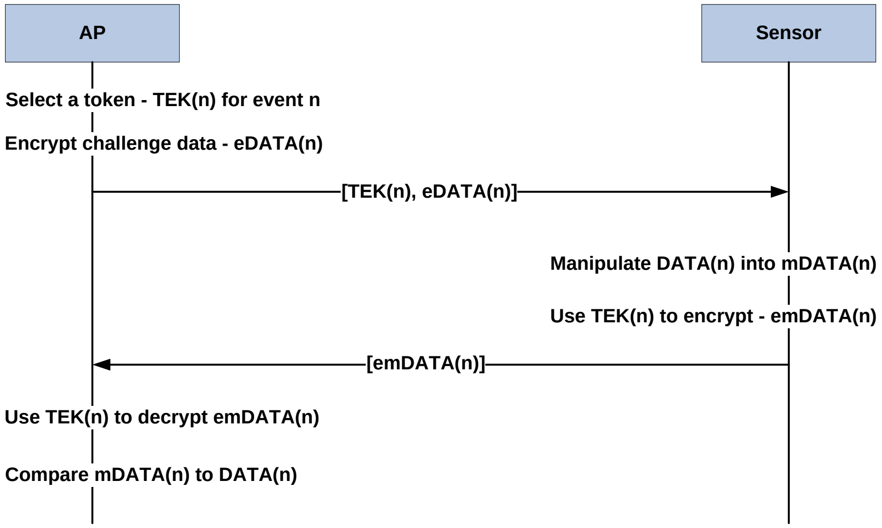

AP
Sensor
Select a token - TEK(n) for event n
Encrypt challenge data - eDATA(n)
[TEK(n), eDATA(n)]
Manipulate DATA(n) into mDATA(n)
Use TEK(n) to encrypt - emDATA(n)
[emDATA(n)]
Use TEK(n) to decrypt emDATA(n)
Compare mDATA(n) to DATA(n)

Figure 4: Access Point authenticates a Sensor

TEK implementation

The details of TEK implementation are covered in US patent 9,806,888 published on October 31 2017.
Figure 3 shows the hardware part of TEK solution. TEK logic includes a method to convert a token to a key. To accomplish this task, a dedicated storage area is written with random values. TEK logic uses a token to assemble an encryption key from the stored random values inside the security HW module. Security keys and the stored random value are not accessible outside the module. The rest of the circuitry is similar to Figure 2.

Figure 4 depicts an Access Point authenticating a Sensor using two messages. The first message from the Access point to the Sensor sends encrypted data and TEK. The Sensor processes that data based on that TEK and sends a message with encrypted data to the Access Point. This challenge/response type handshake may be repeated continuously thanks to low overhead and short latency.

Better protection against man-in-the-middle attack requires mutual authentication. In this case, the Sensor in Figure 4 will initiate a challenge/response type handshake with the Access Point before returning the Figure 4 response to the Access Point. Mutual authentication is required for correct management of TEKs.

End to End authentication and Data Protection is accomplished by associating the random stored values with a Device Identification. The association is done during manufacturing and can be recorded in a data base available to a server for a complete solution. Partial sharing of that data base with, for example, an Access Point, provides quick and low overhead protection.

Possible use of Tokenized Keys

The fact that each and every wireless packet may be encrypted with different key in IoT scenarios has many implications. The sequence of Figure 4 can be used to authenticate an electronic payment transaction with smart credit card or smart phone. Full authentication is accomplished automatically, without user intervention and with minimal overhead.

TEK technology enables the "Hands free" provisioning of networked devices. Activities such as Bluetooth© Mesh commissioning are eliminated - all what the user needs to do is to place the device in its desired location! Home owner standing on a ladder will add Bluetooth© enabled light bulb to the mesh by carrying a phone in pocket and screwing the bulb to its socket.

Smart car with Bluetooth© connectivity and TEK technology will enable user with phone in pocket to enter and start that car. The car manufacturer may supply the user with TEK enabled keys that accomplish the same and are un-clone able.

The added level of protection offered by TEK is useful in case of critical applications having sensors and actuators that are continuously communicating wirelessly. Today's top protection can be enhanced with TEK to protect from future advances in the ability to compromise data.

The low overhead associated with TEK may find new possibilities such as "continuous authentication" where connected devices are verified frequently and not only during initial configuration as often is the case.

Some potential markets for TEK include smart phones, IoT Edge Devices, critical sensors and actuators, car "on-board" computers, car keys, high value medication and other asset tracking.

For more information email shimon@DeviceSYS.com