webdyn
Flexitron group

WebdynEasy W M-Bus

Application Note

AES128 CBC Encryption

Table of contents

1 AES-128 Encryption of BSON Files

1.1 Subject

This implementation note describes the encryption mechanism for BSON files used by WebdynEasy Wireless M-Bus.

It is intended for system integrators who want to generate or parse configuration files compatible with the hub.

1.2 Principle

The hub can protect BSON files using AES encryption to ensure their confidentiality.

> **Warning**
>
> Starting with Firmware V4.02, encryption is enabled on all gateways using a unique key for each product.

When encryption is enabled:

- All BSON files are encrypted using AES-128 in CBC mode;
- The encrypted file is given the extension `.aes`.

For example:

BSON File	Encrypted file
`123456-config.bson`	`123456-config.bson.aes`

1.3 Settings

Parameter	Value
Algorithm	AES-128
Fashion	CBC
Key size	128 bits (16 bytes)
Initialization Vector (IV) Size	16 bytes
Padding	PKCS#7 (PKCS5Padding in some libraries)
Encoding the key in the JSON file	Base64

The AES key is entered in the field:

`config.security.key`

in Base64 format.

Example:

```
Hex :
44:00:AA:F7:83:78:04:9C:AB:13:EE:4E:35:0E:28:1B

Base64 :
RACq94N4BJyrE+5ONQ4oGw==
```

1.4 Firmware V4.0 and later

Starting with version V4.0:

- A random IV is generated for each encryption operation;
- The IV is appended to the end of the encrypted file;
- The IV must be different for each generation in order to comply with cryptographic best practices.

The parameter:

`config.security.aesLegacy`

allows the system to retain its historical behavior (fixed IV) when necessary to ensure compatibility with older firmware versions.

1.5 Encrypted file format

For firmware versions V4.0 and later, the file consists of:

```
+----------------------+------------------+
| Ciphertext           | IV (16 octets)   |
+----------------------+------------------+
```

where:

The ciphertext is the result of AES-128-CBC encryption; IV corresponds to the last 16 bytes of the file; The IV is stored as binary data.

1.6 Decryption Procedure

1.6.1 For firmware versions V4.0 and later:

- Read the last 16 bytes of the file.
- Use these 16 bytes as the IV.
- Treat the rest of the file as ciphertext.
- Decrypt using AES-128-CBC.
- Remove the PKCS#7 padding.
- The result matches the original BSON file.

1.6.2 For firmware versions earlier than V4.0:

- use the fixed IV defined in §4;
- Treat the entire file as the ciphertext.

1.7 Configuration Example

```
{
  "config": {
    "security": {
      "crcMode": 1,
```

```
        "encryption": true,
        "key": {
          "$type": "00",
          "$binary": "RACq94N4BJyrE+5ONQ4oGw=="
        }
      }
    },
    "crc": 0
}
```