webdyn
Flexitron group

WebdynSunPM
Application note

SMS Encryption - xen2 Format

Table of contents

1 Purpose 3
2 SMS Format 3
3 Encryption Inputs 3
4 Key Derivation 3
5 Encryption 3
6 Payload Construction 4
7 SMS Splitting 4
8 Critical Interoperability Constraints 4

Doc ID	SMS-ENC-xen2	**Version**	1
Lifecycle	released	**Updated**	2026-01-28

SMS Encryption - `xen2` Format

1. Purpose

This document specifies the **WebdynSunPM SMS encryption format** for integrators, so that encrypted SMS messages can be sent and successfully decrypted by the product.

2. SMS Format

Each SMS must follow this format:

```
xen2<i>/<N><DATA_BASE64>
```

Where:

- `xen2` : format identifier / version
- `i` : part index (1-based)
- `N` : total number of parts (max 9)
- `<DATA_BASE64>` : fragment of the Base64-encoded payload

No separator (space, `:`, etc.) must be inserted between the header and the data.

3. Encryption Inputs

The integrator provides:

- **Message**: UTF-8 encoded plaintext
- **Password**: UTF-8 string shared with the product

The password is stored on the product side and used to derive the decryption key.

4. Key Derivation

The AES key must be derived from the password using PBKDF2.

Parameters:

- Algorithm: `PBKDF2-HMAC-SHA256`
- Iterations: `100000`
- Salt: `16 bytes`, random
- Key length: `32 bytes` (AES-256)

5. Encryption

The message must be encrypted with:

- Algorithm: **AES-256-GCM**
- Key: derived via PBKDF2
- Nonce: `12 bytes`, random
- AAD: **none** (null / empty)
- Authentication tag: `16 bytes`

The plaintext is UTF-8 encoded prior to encryption.

6. Payload Construction

The binary payload must be assembled in this exact order:

```
payload =
    salt (16 bytes)
 || nonce (12 bytes)
 || ciphertext
 || tag (16 bytes)
```

This `payload` is then encoded using **standard Base64**:

- Alphabet: `A-Z a-z 0-9 + /`
- Padding: `=` allowed and expected

7. SMS Splitting

- Maximum length per SMS: **160 characters**
- Maximum number of parts: **9**
- Splitting is performed on the **Base64 string**

For each part:

```
HEADER = "xen2<i>/<N>"
MAX_DATA_LEN = 160 - len(HEADER)
SMS = HEADER + BASE64[offset : offset + MAX_DATA_LEN]
```

Parts must be sent in ascending order.

8. Critical Interoperability Constraints

The integrator MUST comply with:

- Strict UTF-8 (message and password)
- PBKDF2 with **100000 iterations**, SHA-256
- AES-GCM **without AAD**
- `12-byte` nonce
- `16-byte` tag
- Strict payload ordering
- **Standard** Base64 (not Base64URL)

> **Warning**
>
> Any deviation makes the message impossible to decrypt.