Guidelines for Encrypting Supporting Documents of Claims Submitted via the PhilHealth E-Claims Web Service (PECWS)

Procedures

1. **File Format and Requirements**
    - Supporting documents for e-claims must be in **PDF** (e.g., scanned Claim Signature Form (CSF)) or **XML format** (e.g., Claim Form 4 (CF4)).
    - Scanned documents should comply with the PDF/A standard for compatibility and long-term archiving.
2. **Individual File Encryption**
    - Each file must be encrypted **individually**.
    - Encryption will be performed by the system of the **Health Facility (HF). PECWS** does not provide a service or method for encryption of the e-claim attachments.
3. **File Hashing**
    - Before encryption, compute the file's hash using the **SHA-256 algorithm**.
    - The hash ensures data integrity, verifying that the file remains unchanged after encryption and decryption.
    - On decryption, PhilHealth will recompute the hash to confirm the file's integrity.
4. **Encryption Methodology**
    - Files will be encrypted using **AES-256-CBC**.
    - To decrypt the file, a password is required. The password must also be encrypted using **public key encryption**.
    - PhilHealth will provide a public key (via a digital certificate or file) for encrypting the password.
5. **Encryption Tools**
    - HCIs/SPs may use any preferred programming language or tool to implement encryption.
6. **Password Requirements**
    - Generate a **32-byte random password**:
        - Create two random arrays of **16 bytes each**.
        - Concatenate these arrays to form the 32-byte password.
    - Encrypt each 16-byte array separately using the public key provided by PhilHealth.
7. **Initialization Vector (IV)**
    - Generate a **random 16-byte array** (128 bits) for the Initialization Vector (IV).
    - Encrypt the IV using the provided public key provided by PhilHealth.
8. **Output File Structure**
    - The encrypted file may be renamed using the original file name followed by **".enc" extension**.
    - Encode encrypted data elements using **Base64 encoding**.
    - All metadata and data elements will be combined in a single **JSON file**.

Output File Format

The output file will include both encrypted and unencrypted data elements, structured as follows:

```
{
  "docMimeType": "{MIME type of the attachment file, e.g., 'application/pdf'}",
  "hash": "{SHA-256 hash of the attachment file before encryption}",
  "key1": "{Base64 encoded, public key-encrypted first 16 bytes of the password}",
  "key2": "{Base64 encoded, public key-encrypted second 16 bytes of the password}",
  "iv": "{Base64 encoded, public key-encrypted initialization vector}",
  "doc": "{Base64 encoded, AES-256-CBC encrypted attachment file data}"
}
```

Additional Notes

- All encrypted data elements (keys, IV, and document data) must be securely encoded in Base64 format for transport.
- Ensure that all encryption steps are performed securely to prevent unauthorized access to sensitive claim information.