

Аналіз останньої версії проекту Технічних специфікацій форматів криптографічних повідомлень (захищені дані), розроблених ДССЗІ України, показав, що є ряд принципів зауважень (стаття [«К вопросу реализации стандартов Diffie-Hellman в технических спецификациях Национальной системы электронной подписи Украины»](#)) щодо підходів та описів реалізації алгоритмів у цих специфікаціях. Тому, після обговорення проекту на спільній нараді у Держкомпідприємстві 5.02.2010р. та 16.02.2010 р., було вирішено, що спеціалісти "АМВ group" розроблять та нададуть для обговорення альтернативну редакцію вказаного проекту Технічних специфікацій. Зазначена альтернативна редакція Технічних специфікацій форматів криптографічних повідомлень (захищені дані) і надається далі у цьому документі.

Зауваження та пропозиції щодо цього проекту Технічних специфікацій просимо направляти на електронну адресу автора, вказану на завершення цього документа.

---

## ТЕХНІЧНІ СПЕЦИФІКАЦІЇ форматів криптографічних повідомлень. Захищені дані

### I. Загальні положення та визначення термінів

1.1. Технічні специфікації форматів криптографічних повідомлень (далі – Технічні специфікації) визначають синтаксис (формат представлення) зашифрованого повідомлення (даних) в електронній формі, а також протоколи управління ключами, які повинні застосовуватися для цього синтаксису з метою узгодження ключів. Встановлення єдиних форматів криптографічних повідомлень має на меті визначення технічних умов щодо забезпечення сумісності засобів криптографічного захисту інформації різних розробників.

1.2. У цих Технічних специфікаціях терміни вживаються у такому значенні:  
повідомлення „захищені дані” – повідомлення, що містить цифровий конверт;

цифровий конверт (“enveloped-data”) – зашифровані дані типу „дані” („data”), або „підписані дані” („signed-data”) разом із зашифрованим симетричним ключем;

симетричний ключ сеансу або ключ шифрування даних (КШД) – ключ сеансу, на якому здійснюється шифрування даних за алгоритмом, визначеним у національному стандарті України ДСТУ ГОСТ 28147-2009;

узгоджений ключ („key agreement”) або ключ шифрування ключа (КШК) – симетричний ключ, на якому здійснюється шифрування симетричного ключа сеансу;

протокол управління ключами (або протокол узгодження ключа) – протокол Діффі-Геллмана обчислення КШК в циклічній групі простого поля або в групі точок еліптичної кривої;

механізмі узгодження ключа – мається на увазі статичний (Static-Static mode) або динамічний (Ephemeral-Static mode) механізм узгодження ключа, визначені у цих Технічних специфікаціях;

дані – повідомлення або частина повідомлення, які не оброблюють і не змінюють в процесі обробки.

Інші терміни вживаються у значенні, наведеному у Законі України “Про електронний цифровий підпис”, Порядку акредитації центру сертифікації ключів, затвердженому постановою Кабінету Міністрів України від 13 липня 2004 року № 903, Правилах посиленої сертифікації, затверджених наказом Департаменту спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України № 3 від 13.01.2005, зареєстрованих в Міністерстві юстиції України 27.01.2005 за № 104/10384, інших нормативно-правових актах з питань криптографічного та технічного захисту інформації.

Скорочення та позначення:

CMS – синтаксис криптографічного повідомлення (Cryptographic Message Syntax).

DH - протокол Діффі-Геллмана (Diffie-Hellman) в циклічній групі простого поля; може використовуватися також позначення FFC DH (Finite Field Cryptography Diffie-Hellman) - протокол Діффі-Геллмана в криптографії скінченного поля.

ECDH - протокол Діффі-Геллмана (Diffie-Hellman) в групі точок еліптичної кривої; може використовуватися також позначення ECC DH (Elliptic Curve Cryptography Diffie-Hellman) - протокол Діффі-Геллмана в криптографії з еліптичними кривими).

ДКЕ - довгостроковий ключовий елемент.

### 1.3. Технічні специфікації засновані на

міжнародних рекомендаціях:

RFC 2631 “Diffie-Hellman Key Agreement Method”, June 1999,

RFC 2785 Methods for Avoiding the "Small-Subgroup" Attacks on the Diffie-Hellman Key Agreement for S/MIME, March 2000,

RFC 3279 “Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”, April 2002,

RFC 3281 “An Internet Attribute Certificate Profile for Authorization”, April 2002,

RFC 3370 “Cryptographic Message Syntax (CMS) Algorithms”, August 2002,

RFC 3394 “Encryption Standard (AES) Key Wrap Algorithm”, September 2002,

RFC 3852 “Cryptographic Message Syntax (CMS)”, July 2004,

RFC 4490 - Using the GOST 28147-89, GOST R 34.11-94, GOST R 34.10-94, and GOST R 34.10-2001 Algorithms with Cryptographic Message Syntax (CMS), May 2006,

RFC 5008 “Suite B in Secure/Multipurpose Internet Mail Extensions (S/MIME)”, September 2007,

RFC 5480 “Elliptic Curve Cryptography Subject Public Key”, March 2009,

RFC 5652 “Cryptographic Message Syntax (CMS)”, September 2009;

національних стандартах України:

ДСТУ 4145-2002 “Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевіряння”,

ДСТУ ISO/IEC 11770-3:2002 “Інформаційні технології. Методи захисту. Управління ключовими даними. Частина 3. Протоколи, що ґрунтуються на асиметричних криптографічних перетвореннях”,

ДСТУ ISO/IEC 15946-3:2002 “Інформаційні технології. Методи захисту. Криптографічні методи, що ґрунтуються на еліптичних кривих. Частина 3. Установлення ключів”,

ДСТУ ГОСТ 28147-89 “Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования”,

ДСТУ ISO/IEC 10118-3:2005 “Інформаційні технології. Методи захисту. Геш-функції. Частина 3. Спеціалізовані геш-функції”;

міждержавних стандартах:

ГОСТ 34.310-95 “Информационные технологии. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма”,

ГОСТ 34.311-95 “Информационная технология. Криптографическая защита информации. Функция хеширования”;

додатку 1 до Інструкції про порядок постачання і використання ключів до засобів криптографічного захисту інформації, затвердженої наказом Адміністрації Державної служби спеціального зв'язку та захисту інформації України № 114 від 12.06.2007 та зареєстрованої в Міністерстві юстиції України 25.06.2007 за № 729/13996 (надалі “Інструкція №114”),

та Технічних специфікаціях форматів представлення базових об'єктів національної системи електронного цифрового підпису, затверджених наказом Департаменту спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України, Державного департаменту з питань зв'язку та інформатизації Міністерства транспорту та зв'язку України від 11.09.2006 № 99 /166 (надалі – “Формати представлення базових об'єктів”).

1.4. Якщо в Технічних специфікаціях існують розходження із нормативно-правовими актами та нормативними документами, зазначеними у пункті 1.3

Технічних специфікацій, то використовуються положення цих Технічних специфікацій.

## II. Типи повідомлень

2.1. Технічні специфікації визначають тип інформаційного повідомлення “ContentInfo”, що включає в себе один певний тип даних, який в свою чергу може також включати дані певного типу.

2.2. Тип повідомлення “ContentInfo” представлено у нотації ASN.1, визначеній у міжнародному стандарті ISO/IEC 8824 “Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)”.

2.3. Усі структури даних кодуються за правилами DER згідно з міжнародними стандартами ISO/IEC 8825-1:2002 “Information technology – ASN.1 encoding Rules – Part 1: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)” та AMD1:2004 “Support for EX-TENDED-XER”.

### 2.4. Формат повідомлення “ContentInfo”

2.4.1. На тип “ContentInfo” вказує такий об’єктний ідентифікатор:

```
id-ct-contentInfo OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
rsadsi(113549) pkcs(1) pkcs9(9) smime(16) ct(1) 6 }
```

2.4.2. Інформаційне повідомлення “ContentInfo” має такий формат:

```
ContentInfo ::= SEQUENCE {
    contentType          ContentType,
    content              [0] EXPLICIT ANY DEFINED BY contentType }
```

```
ContentType ::= OBJECT IDENTIFIER
```

2.4.3. Поля структури “ContentInfo” мають такі значення:

“contentType” – об’єктний ідентифікатор, що вказує на тип пов’язаних з ним даних, наприклад тип “захищені дані”;

“Content” – пов’язані з об’єктним ідентифікатором дані. Тип даних однозначно визначається полем “contentType”.

2.5. Повідомлення, що містить цифровий конверт, має тип даних “enveloped-data” (“захищені дані”). Повідомлення “захищені дані” включається в повідомлення інформаційного типу “ContentInfo”.

Об’єктний ідентифікатор

id-envelopedData OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs7(7) 3 }

вказує на те, що структура “ContentInfo” містить дані типу “захищені дані”.

2.6. Повідомлення “захищені дані” включає в себе інші типи повідомлень, а саме: “дані” (“data”) або “підписані дані” (“signed-data”).

При включенні в повідомлення “захищені дані” повідомлення типу “дані” автентифікація відправника цих даних не забезпечується, якщо використовується динамічний механізм узгодження ключів (див. пункт 3.3). При включенні в повідомлення “захищені дані” повідомлення типу “підписані дані” завжди забезпечується автентифікація відправника цих даних.

2.7. Формат повідомлення “підписані дані” (“signed-data”) встановлюється технічними специфікаціями форматів підписаних електронних даних.

На тип “ signed-data” вказує такий об’єктний ідентифікатор:

id-signedData OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs7(7) 2 }

2.8. Повідомлення типу “дані” призначено для представлення довільних рядків октетів, наприклад текстових файлів ASCII. Інтерпретація таких даних покладається на програмний додаток.

На тип “data” вказує такий об’єктний ідентифікатор:

id-data OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs7(7) 1 }

### **III. Процедура формування та розкриття “захищених даних”**

#### **3.1. Процедура формування “захищених даних” відправником**

3.1.1. Відправник за протоколом управління ключами за допомогою особистого ключа відправника та відкритого ключа одержувача формує узгоджений ключ, на якому шифрує симетричний ключ сеансу КШД.

3.1.2. Зашифрований симетричний ключ сеансу КШД та інша інформація для одержувача включається в структуру “RecipientInfo” (інформація одержувача). Структура “RecipientInfo” наводиться у пункті 4.4 Технічних специфікацій.

3.1.3. Дані зашифровуються на симетричному ключі сеансу КШД.

3.1.4. Структура “RecipientInfo” разом із зашифрованими даними вкладається у структуру “enveloped-data”. Структура “enveloped-data” наводиться у пункті 4.1 Технічних специфікацій.

3.1.5. Зашифровані дані розміщуються у полі “EnvelopedData encryptedContentInfo encryptedContent OCTET STRING” структури “enveloped-data”.

3.2. Процедура розкриття “захищених даних” одержувачем

3.2.1. Одержувач за протоколом управління ключами за допомогою відкритого ключа відправника та особистого ключа одержувача формує узгоджений ключ, на якому розшифровує симетричний ключ сеансу КШД.

Інформація для одержувача, яка необхідна для реалізації протоколу управління ключами з боку одержувача, а також для розшифрування повідомлення подається в структурі “RecipientInfo”. Структура “RecipientInfo” наводиться у пункті 4.4 Технічних специфікацій.

3.2.2. Одержувач за допомогою симетричного ключа сеансу КШД розшифровує дані, використовуючи алгоритм шифрування, визначений у структурі “RecipientInfo”.

3.3. Особливості формування повідомлення “захищені дані”

3.3.1. Засоби криптографічного захисту інформації відправника та одержувача повинні підтримувати криптографічні алгоритми, визначені цими Технічними специфікаціями.

3.3.2. При використанні криптографічних перетворень в циклічній групі простого поля та в групі точок еліптичної кривої застосовуються статичний та динамічний механізми узгодження ключів.

3.3.2.1. Статичний механізм узгодження ключів (“Static-Static mode”) – узгодження ключів типу Діффі-Геллмана, при якому як відправник, так і одержувач мають статичну, засвідчену сертифікатом X.509, ключову пару. Тим самим цей статичний механізм забезпечує автентифікацію відправника повідомлення типу “захищені дані”.

Статичний механізм узгодження ключів може використовуватися лише у випадку, коли параметри криптографічного алгоритму статичної ключової пари відправника еквівалентні параметрам криптографічного алгоритму статичної ключової пари одержувача. Якщо зазначені параметри не еквівалентні, повинен застосовуватися динамічний механізм узгодження ключів.

3.3.2.2. При статичному механізмі узгодження ключа для формування узгодженого ключа відправник повинен використовувати особистий асиметричний ключ відправника та відкритий ключ одержувача, одержувач повинен використовувати особистий асиметричний ключ одержувача та відкритий ключ відправника.

Відкриті ключі відправника та одержувача обираються з сертифікатів відкритих ключів (сертифікатів шифрування).

3.3.2.3. Динамічний механізм узгодження ключів (“Ephemeral-Static mode”) – узгодження ключів типу Діффі-Геллмана, при якому одержувач має статичну, засвідчену сертифікатом X.509, ключову пару, а відправник генерує нову (сеансову/ динамічну) ключову пару для кожного повідомлення і посилає відкритий ключ цієї пари одержувачу, використовуючи “originatorKey” структури “RecipientInfo”.

При цьому параметри криптографічного алгоритму динамічної ключової пари відправника повинні бути еквівалентні параметрам криптографічного алгоритму статичної ключової пари одержувача.

3.3.2.4. При динамічному механізмі узгодження ключа для формування узгодженого ключа відправник повинен використовувати особистий асиметричний ключ сеансу (маркер), що генерується відправником під час формування захищених даних, та відкритий довгостроковий ключ одержувача; одержувач повинен використовувати особистий довгостроковий ключ та відкритий сеансовий ключ відправника (маркер), що отримується від відправника в кожному сеансі у полі “originatorKey” структури “RecipientInfo”.

3.3.4. Особливості кодування параметрів криптографічного алгоритму визначено у пункті 4.5.4.4 Технічних специфікацій.

3.3.5. Протокол Діффі-Геллмана в циклічній групі простого поля (DH) використовується для ключових пар (відправника та одержувача), що відповідають ГОСТ 34.310-95.

3.3.6. Протокол Діффі-Геллмана в групі точок еліптичної кривої (ECDH) використовується для ключових пар (відправника та одержувача), що відповідають ДСТУ 4145-2002.

## IV. Представлення структури “захищені дані”

### 4.1. Формат структури “EnvelopedData”

Структура “захищені дані” має такий формат [RFC 3852/ RFC 5652]:

```
EnvelopedData ::= SEQUENCE {  
    version                CMSVersion,  
    originatorInfo         [0] IMPLICIT OriginatorInfo OPTIONAL,  
    recipientInfos         RecipientInfos,  
    encryptedContentInfo   EncryptedContentInfo,  
    unprotectedAttrs      [1] IMPLICIT UnprotectedAttributes OPTIONAL }
```

```
CMSVersion ::= INTEGER { v0(0), v1(1), v2(2), v3(3), v4(4), v5(5) }
```

```
OriginatorInfo ::= SEQUENCE {  
    certificates           [0] CertificateSet OPTIONAL,  
    crls                   [1] CertificateRevocationLists OPTIONAL }
```

```
RecipientInfos ::= SET SIZE (1..MAX) OF RecipientInfo
```

```
EncryptedContentInfo ::= SEQUENCE {  
    contentType           ContentType,  
    contentEncryptionAlgorithm ContentEncryptionAlgorithmIdentifier,  
    encryptedContent      [0] IMPLICIT EncryptedContent OPTIONAL }
```

```
EncryptedContent ::= OCTET STRING
```

```
UnprotectedAttributes ::= SET SIZE (1..MAX) OF Attribute
```

Звичайно типове застосування “захищені дані” містить одного або більше одержувачів цифрового конверту, що містить “дані” або “підписані дані”.

Процедура формування “захищені дані” містить такі кроки:

1. Генерується випадково симетричний ключ сеансу КШД.
2. Використовуючи статичний чи динамічний механізм узгодження ключів обчислюється ключ шифрування ключа КШК для кожного одержувача.
3. Симетричний ключ сеансу КШД шифрується на ключі шифрування ключа КШК для кожного одержувача.
4. Для кожного одержувача зашифрований ключ КШД і інша відповідна специфічна інформація розміщуються всередині значення RecipientInfo.
5. Значення RecipientInfo для всіх одержувачів, разом з зашифрованими даними, розміщаються в EnvelopedData.

### 4.2. Поля структури “EnvelopedData”



4.2.1. Поле “Version” визначає номер версії синтаксису, який повинен мати значення 2.

4.2.2. Поле “originatorInfo” містить сертифікати відкритих ключів та списки відкликаних сертифікатів відправника. Поле є необов’язковим.

certs - це ланцюжок (chain) сертифікатів відправника, пов’язаний із статичним механізмом узгодження ключів, який було застосовано. certs може містити лише кінцевий сертифікат статичної ключової пари відправника, або може містити повний ланцюжок (chain) сертифікатів, достатній для побудови шляху сертифікації від довіреного “кореня”, або може містити не повний ланцюжок (chain) сертифікатів, наприклад, кінцевий сертифікат відправника та сертифікат його центра сертифікації. Сертифікати розміщуються у такому порядку: першим (з найменшим індексом) розміщується сертифікат центра сертифікації вищого рівня (кореневий для повного ланцюга сертифікатів), останнім (з найбільшим індексом) розміщується сертифікат відправника, який було застосовано для статичного механізму.

Наявність сертифікату відправника робить структуру захищених даних “самодостатньою” у тому розумінні, що дозволяє одержувачу розшифрувати повідомлення, використовуючи відкритий ключ відправника із його сертифіката в полі “originatorInfo” без необхідності пошуку сертифіката відправника у сховищі сертифікатів.

Поле “certs” має такий формат [RFC 3852/ RFC 5652]:

```
CertificateSet ::= SET OF CertificateChoices
```

```
CertificateChoices ::= CHOICE {  
    certificate Certificate,  
    v2AttrCert [2] IMPLICIT AttributeCertificateV2,  
    other [3] IMPLICIT OtherCertificateFormat }
```

```
OtherCertificateFormat ::= SEQUENCE {  
    otherCertFormat OBJECT IDENTIFIER,  
    otherCert ANY DEFINED BY otherCertFormat }
```

```
AttributeCertificateV2 ::= AttributeCertificate
```

Поле “v2AttrCert” (“AttributeCertificate”) має такий формат [RFC 3281]:

```
AttributeCertificate ::= SEQUENCE {  
    acinfo AttributeCertificateInfo,  
    signatureAlgorithm AlgorithmIdentifier,  
    signatureValue BIT STRING  
}
```

```

AttributeCertificateInfo ::= SEQUENCE {
    version          AttCertVersion -- version is v2,
    holder           Holder,
    issuer           AttCertIssuer,
    signature        AlgorithmIdentifier,
    serialNumber     CertificateSerialNumber,
    attrCertValidityPeriod AttCertValidityPeriod,
    attributes       SEQUENCE OF Attribute,
    issuerUniqueID   UniqueIdentifier OPTIONAL,
    extensions       Extensions OPTIONAL
}

AttCertVersion ::= INTEGER { v2(1) }

Holder ::= SEQUENCE {
    baseCertificateID [0] IssuerSerial OPTIONAL,
        -- the issuer and serial number of
        -- the holder's Public Key Certificate

    entityName       [1] GeneralNames OPTIONAL,
        -- the name of the claimant or role

    objectDigestInfo [2] ObjectDigestInfo OPTIONAL
        -- used to directly authenticate the holder,
        -- for example, an executable
}

ObjectDigestInfo ::= SEQUENCE {
    digestedObjectType ENUMERATED {
    publicKey          (0),
    publicKeyCert     (1),
    otherObjectTypes (2) },
        -- otherObjectTypes MUST NOT
        -- be used in this profile

    otherObjectTypeID OBJECT IDENTIFIER OPTIONAL,
    digestAlgorithm    AlgorithmIdentifier,
    objectDigest       BIT STRING
}

AttCertIssuer ::= CHOICE {
    v1Form GeneralNames, -- MUST NOT be used in this
        -- profile
    v2Form [0] V2Form -- v2 only
}

```

```
V2Form ::= SEQUENCE {
    issuerName      GeneralNames OPTIONAL,
    baseCertificateID [0] IssuerSerial OPTIONAL,
    objectDigestInfo [1] ObjectDigestInfo OPTIONAL
    -- issuerName MUST be present in this profile
    -- baseCertificateID and objectDigestInfo MUST NOT
    -- be present in this profile
}
```

```
IssuerSerial ::= SEQUENCE {
    issuer      GeneralNames,
    serial      CertificateSerialNumber,
    issuerUID   UniqueIdentifier OPTIONAL
}
```

```
AttCertValidityPeriod ::= SEQUENCE {
    notBeforeTime GeneralizedTime,
    notAfterTime  GeneralizedTime
}
```

crls - це колекція списків відкликання (CRL). Набір CRL містить інформацію, достатню для того, щоб визначити чи є сертифікати в полі certs чинними. Послідовність розміщення CRL в колекції повинна відповідати послідовності розміщення сертифікатів у колекції certs.

Наявність списків відкликання дозволяє одержувачу визначити чинність сертифіката відправника на момент формування захищених даних без необхідності звернення до зовнішніх джерел розміщення CRL.

Поле “crls” має такий формат [RFC 3852/ RFC 5652]:

```
RevocationInfoChoices ::= SET OF RevocationInfoChoice
```

```
RevocationInfoChoice ::= CHOICE {
    crl CertificateList,
    other [1] IMPLICIT OtherRevocationInfoFormat }
```

```
OtherRevocationInfoFormat ::= SEQUENCE {
    otherRevInfoFormat OBJECT IDENTIFIER,
    otherRevInfo ANY DEFINED BY otherRevInfoFormat }
```

“CertificateList” може містити CRL, або частковий Delta CRL, формат яких визначено у Технічній специфікації “Форматах представлення базових об’єктів”.

4.2.3. Поле “recipientInfos” містить набір інформації для одержувача.

recipientInfos - це колекція інформації про кожного із одержувачів. Колекція може мати більш ніж один елемент.

4.2.4. Поле “encryptedContentInfo” містить зашифроване повідомлення.

4.2.5. Поле “Unprotected Attrs” містить набір атрибутів, що не шифруються разом з повідомленням. Зазначене поле не використовується.

4.3. Поля структури “EncryptedContentInfo”

4.3.1. Поле “contentType” вказує на тип даних.

4.3.2. Поле “contentEncryptionAlgorithm” визначає криптографічний алгоритм шифрування даних. Для всіх одержувачів повідомлення повинен застосовуватися однаковий алгоритм шифрування даних та однаковий ключ шифрування даних з параметрами алгоритму.

4.3.4. Поле “encryptedContent” містить дані, зашифровані з використанням алгоритму, що визначено у полі “contentEncryptionAlgorithm” та ключа шифрування даних КШД. Поле є необов’язковим. У разі відсутності поля “encryptedContent” вважається, що зашифровані дані представляються у інший спосіб.

4.4. Структура “RecipientInfo” має такий формат:

```
RecipientInfo ::= CHOICE {  
  kari [1] KeyAgreeRecipientInfo }
```

Тип “KeyAgreeRecipientInfo” призначений для кодування даних, що використовуються одержувачем у протоколі управління ключами.

4.4.1. Структура “KeyAgreeRecipientInfo” має такий формат:

```
KeyAgreeRecipientInfo ::= SEQUENCE {  
  version CMSVersion,  
  originator [0] EXPLICIT OriginatorIdentifierOrKey,  
  ukm [1] EXPLICIT UserKeyingMaterial OPTIONAL,  
  keyEncryptionAlgorithm KeyEncryptionAlgorithmIdentifier,  
  recipientEncryptedKeys RecipientEncryptedKeys }
```

```
OriginatorIdentifierOrKey ::= CHOICE {  
  issuerAndSerialNumber IssuerAndSerialNumber,  
  subjectKeyIdentifier [0] SubjectKeyIdentifier,  
  originatorKey [1] OriginatorPublicKey }
```

OriginatorPublicKey ::= SEQUENCE {  
    algorithm        AlgorithmIdentifier,  
    publicKey        BIT STRING}

UserKeyingMaterial ::= OCTET STRING

KeyEncryptionAlgorithmIdentifier ::= AlgorithmIdentifier

AlgorithmIdentifier ::= SEQUENCE {  
    algorithm        OBJECT IDENTIFIER,  
    parameters      ANY DEFINED BY algorithm}

RecipientEncryptedKeys ::= SEQUENCE OF RecipientEncryptedKey

RecipientEncryptedKey ::= SEQUENCE {  
    rid              KeyAgreeRecipientIdentifier,  
    encryptedKey     EncryptedKey}

EncryptedKey ::= OCTET STRING

KeyAgreeRecipientIdentifier ::= CHOICE {  
    issuerAndSerialNumber IssuerAndSerialNumber,  
    rKeyId                [0] IMPLICIT RecipientKeyId}

IssuerAndSerialNumber ::= SEQUENCE {  
    issuer            Name,  
    serialNumber     CertificateSerialNumber}

CertificateSerialNumber ::= INTEGER

#### 4.4.2. Поля структури “KeyAgreeRecipientInfo”

4.4.2.1. Поле “Version” визначає номер версії синтаксису, який повинен мати значення 3.

4.4.2.2. Поле “originator” містить ідентифікаційні дані відправника. Тип цих даних залежить від використаного механізму (протоколу) узгодження ключа.

4.4.2.2.1. При застосуванні статичного механізму узгодження ключів типу Діффі-Геллмана в якості ідентифікатора відправника повинні використовуватися ім'я емітента сертифікату (центра сертифікації) та серійний номер сертифікату відкритого ключа відправника “issuerAndSerialNumber” або ідентифікатор ключа суб'єкта сертифікату відкритого ключа відправника “subjectKeyId”.

4.4.2.2.2. При застосуванні динамічного механізму узгодження ключів типу Діффі-Геллмана в якості ідентифікаційних даних відправника застосовується його відкритий асиметричний ключ сеансу (маркер), що генерується відправником, “originatorKey”.

4.4.2.2.3. При застосуванні динамічного механізму узгодження ключів в циклічній групі простого поля (DH) поле “algorithm” в “originatorKey” повинно містити ідентифікатор відкритого ключа ГОСТ 34.310:

```
id-gost34310 OBJECT IDENTIFIER ::= { iso(1) member-body(2) ukraine(804) root(2) security(1) cryptography(1) pki(1) pki-alg(1) pki-alg-asym(3) gost34310(2) }
```

Відповідно до RFC 3370, параметри алгоритму поля “algorithm” в “originatorKey” повинні бути відсутні.

Поле “originatorKey publicKey” повинно містити відкритий ключ відправника (маркер), що має такий формат:

PublicKey ::= INTEGER, що інкапсулюється в BIT STRING

Відкритий ключ ГОСТ 34.310-95 кодується як ціле, відповідно до “Форматів представлення базових об’єктів”.

4.4.2.2.4. При застосуванні динамічного механізму узгодження ключів в групі точок еліптичної кривої (ECDH) поле “algorithm” в “originatorKey” повинно містити ідентифікатор відкритого ключа ДСТУ 4145-2002:

Поліноміальний базис, формат Little-Endian:

```
id-dstu4145PB OBJECT IDENTIFIER ::= { iso(1) member-body(2) ukraine(804) root(2) security(1) cryptography(1) pki(1) pki-alg(1) pki-alg-asym(3) dstu4145(1) PB(1) }
```

Поліноміальний базис, формат Big-Endian:

```
id-dstu4145PB-std OBJECT IDENTIFIER ::= { iso(1) member-body(2) ukraine(804) root(2) security(1) cryptography(1) pki(1) pki-alg(1) pki-alg-asym(3) dstu4145(1) PB(1) Special curves(1) DSTU key format(1) }
```

Оптимальний базис, формат Little-Endian:

```
id-dstu4145ONB OBJECT IDENTIFIER ::= { iso(1) member-body(2) ukraine(804) root(2) security(1) cryptography(1) pki(1) pki-alg(1) pki-alg-asym(3) dstu4145(1) ONB(2) }
```

Оптимальний базис, формат Big-Endian:

id-dstu4145ONB-std OBJECT IDENTIFIER ::= { iso(1) member-body(2) ukraine(804) root(2) security(1) cryptography(1) pki(1) pki-alg(1) pki-alg-asym(3) dstu4145(1) ONB(1) Special curves(1) DSTU key format(1) }

Відповідно до RFC 5008, параметри алгоритму поля “algorithm” в “originatorKey” повинні бути ASN.1 NULL.

Поле “originatorKey publicKey” повинно містити відкритий ключ відправника (маркер), що має такий формат:

PublicKey ::= OCTET STRING, що інкапсулюється в BIT STRING

Відкритий ключ ДСТУ 4145 2002, відповідно до “Форматів представлення базових об’єктів” – це послідовність байтів, яка являє собою елемент основного поля (згідно з пунктом 5.3 ДСТУ 4145-2002), який є стиснутим зображенням (згідно з пунктом 6.9 ДСТУ 4145-2002) точки на еліптичній кривій, що відображає відкритий ключ ЕЦП. Розмір зображення в байтах дорівнює  $m/8$  заокруглене до найближчого цілого у більшу сторону.

4.4.2.3. Поле “ukm” (User Keying Material - матеріал щодо ключа користувача) містить додаткову інформацію, яку відправник надає одержувачу під час виконання протоколу узгодження ключа типу Діффі-Геллмана.

Поле “ukm” використовується з метою забезпечення можливості формування різних значень узгоджених ключів у різний час суб’єктами, що використовують ті самі довгострокові асиметричні пари ключів (статичні ключі).

Реалізації цієї Специфікації повинні обробляти KeyAgreeRecipientInfo, що містить поле “ukm”.

У разі застосування механізму узгодження ключів в циклічній групі простого поля (DH) в поле “ukm” вноситься значення “partyAInfo” (кодоване як OCTET STRING, деталі у пункті 5.4.1.1), яке генерується відправником та використовується в структурі “OtherInfo”. Якщо “partyAInfo” задано, то воно повинно містити 512 біт (64 байти).

У разі застосування динамічного механізму узгодження ключів в групі точок еліптичної кривої (ECDH) в поле “ukm” вноситься значення “entityUInfo” (кодоване як OCTET STRING, деталі у пункті 5.4.2.2), яке генерується відправником та використовується в структурі “SharedInfo”. Якщо “entityUInfo” задано, то воно повинно містити 512 біт (64 байти).

4.4.2.4. Поле “keyEncryptionAlgorithm” визначає ідентифікатор алгоритму (протоколу) узгодження ключа (Key Agreement Algorithm).

4.4.2.5. Поле “recipientEncryptedKeys” містить ідентифікатор одержувача та зашифрований ключ КШД для одного або більше одержувачів.

4.4.2.6. Поле “KeyAgreeRecipientIdentifier” є структурою з вибором двох альтернатив, що визначають сертифікат одержувача, а отже і відкритий ключ одержувача, що використовується відправником при генерації узгодженого ключа КШК в протоколі Діффі-Геллмана узгодження ключа:

“issuerAndSerialNumber” альтернатива вказує на сертифікат за розпізнавальним ім'ям на центр сертифікації ключів та серійний номер сертифікату відкритого ключа одержувача.

“rKeyId” альтернатива типу RecipientKeyIdentifier має таке значення:

```
RecipientKeyIdentifier ::= SEQUENCE {  
    subjectKeyIdentifier SubjectKeyIdentifier,  
    date                 GeneralizedTime OPTIONAL,  
    other                OtherKeyAttribute OPTIONAL }
```

```
SubjectKeyIdentifier ::= OCTET STRING
```

“subjectKeyIdentifier” ідентифікує сертифікат одержувача його ідентифікатором ключа, який відповідає розширенню subjectKeyIdentifier X.509 сертифікату;

“date” необов'язкове поле. Якщо присутнє, то визначає дату, на яку наперед надісланий одержувачем УКМ використовувався відправником. Використання цього поля не регламентується у межах цього документу.

“other” необов'язкове поле. Якщо присутнє, то це поле містить додаткову інформацію, що використовується одержувачем для розміщення відкритого ключового матеріалу, що використовується відправником. Використання цього поля не регламентується у межах цього документу.

Реалізації цієї Специфікації повинні підтримувати обидві вищевказані альтернативи для визначення сертифіката одержувача.

4.4.2.7. Поле “encryptedKey” містить симетричний ключ КШД, зашифрований на узгодженому ключі КШК.

4.4.3. Особливості синтаксису структури “KeyAgreeRecipientInfo”

4.4.3.1. Ідентифікатор алгоритму (протоколу) узгодження ключа вказується в полі “EnvelopedData RecipientInfos KeyAgreeRecipientInfo keyEncryptionAlgorithm”.

```
KeyEncryptionAlgorithmIdentifier ::= AlgorithmIdentifier
```



```

AlgorithmIdentifier ::= SEQUENCE {
    algorithm          OBJECT IDENTIFIER,
    parameters        ANY DEFINED BY algorithm}

```

Поле “algorithm” повинно містити об’єктний ідентифікатор одного з алгоритмів узгодження ключа, зазначені нижче, а поле “parameters” повинно містити ідентифікатор алгоритму шифрування ключа КШК (KeyWrapAlgorithm):

```
parameters ::= KeyWrapAlgorithm
```

```
KeyWrapAlgorithm ::= AlgorithmIdentifier
```

4.4.3.2. Об’єктний ідентифікатор алгоритму узгодження ключа визначає:

- ZZ-функцію генерації спільного секрету (ZZ) для визначеного протоколу, а саме: в циклічній групі простого поля (DH) або в групі точок еліптичної кривої (ECDH);

- KDF-функцію (Key Derivation Function), функцію формування, на основі спільного секрету та додаткової інформації, ключа шифрування ключа КШК (KM, KeyingMaterial) для заданого алгоритму “KeyWrapAlgorithm”.

4.4.3.3. У Технічних специфікаціях визначаються такі об’єктні ідентифікатори (OID) алгоритму узгодження ключа у циклічній групі простого поля (DH)

4.4.3.3.1. Алгоритм узгодження ключа у циклічній групі простого поля (DH) з використанням геш-функції ГОСТ 34.311 позначається через ідентифікатор “id-alg-DH-ua” – є обов’язковим алгоритмом, алгоритмом за умовчанням; застосовується як для статичного, так і для динамічного механізму узгодження ключа; при цьому ознакою динамічного механізму є ненульове значення поля “originatorKey”, відповідно до пункту 4.4.2.2.2:

```
id-alg-DH-ua OBJECT IDENTIFIER ::= { iso(1) member-body(2) Ukraine(804)
root(2) security(1) cryptography(1) pki(1) pki-alg(1) pki-alg-asm(3) DH-ua(3) }
```

4.4.3.3.2. Для алгоритмів узгодження ключа у циклічній групі простого поля (DH) з використанням, відповідно до національного стандарту України ДСТУ ISO/IEC 10118-3:2005 та RFC 2631, геш-функції SHA-1, використовуються такі ідентифікатори:

- “id-alg-SSDH” – не обов’язковий, застосовується для статичного механізму узгодження ключа

- “id-alg-ESDH” – не обов’язковий, застосовується для динамічного механізму узгодження ключа

```
id-alg-SSDH OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)
```

rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) alg(3) 10 }

id-alg-ESDH OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840)  
rsadsi(113549) pkcs(1) pkcs-9(9) smime(16) alg(3) 5 }

4.4.3.3.3. Алгоритми узгодження ключа, а саме ZZ-функція та KDF-функція, у циклічній групі простого поля (DH) визначені у розділі 5 цих Технічних специфікацій.

4.4.3.4. У Технічних специфікаціях визначаються такі об'єктні ідентифікатори (OID) алгоритму узгодження ключа в групі точок еліптичної кривої (ECDH)

4.4.3.4.1. З використанням геш-функції ГОСТ 34.311:  
алгоритм з кофакторним множенням

“id-dhSinglePass-cofactorDH-gost34311kdf-scheme” - обов'язковий (з кофактором), алгоритм за умовчанням

алгоритм без кофакторного множення

“id-dhSinglePass-stdDH-gost34311kdf-scheme” - обов'язковий (без кофактора)

Відповідно до ДСТУ ISO/IEC 15946-3:2005 кофакторне множення є одним з методів запобігання методам «атаки малої підгрупи» (деталі атаки наведені у міжнародних рекомендаціях RFC 2785).

id-dhSinglePass-cofactorDH-gost34311kdf-scheme OBJECT IDENTIFIER ::= { iso(1) member-body(2) Ukraine(804) root(2) security(1) cryptography(1) pki(1) pki-  
alg(1) pki-alg-asym(3) dhSinglePass-cofactorDH-gost34311kdf (4) }

id-dhSinglePass-stdDH-gost34311kdf-scheme OBJECT IDENTIFIER ::= { iso(1) member-body(2) Ukraine(804) root(2) security(1) cryptography(1) pki(1) pki-  
alg(1) pki-alg-asym(3) dhSinglePass-stdDH-gost34311kdf (5) }

4.4.3.4.2. З використанням, відповідно до національного стандарту України ДСТУ ISO/IEC 15946-3:2005, ДСТУ ISO/IEC 10118-3:2005 та міжнародних рекомендацій RFC 5008, геш-функцій SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 (не обов'язкові):

алгоритми з кофакторним множенням

“id-dhSinglePass-cofactorDH-sha1kdf-scheme”

“id-dhSinglePass-cofactorDH-sha224kdf-scheme”

“id-dhSinglePass-cofactorDH-sha256kdf-scheme”

“id-dhSinglePass-cofactorDH-sha384kdf-scheme”

“id-dhSinglePass-cofactorDH-sha512kdf-scheme”

алгоритми без кофакторного множення

“id-dhSinglePass-stdDH-sha1kdf-scheme”

“id-dhSinglePass-stdDH-sha224kdf-scheme”  
“id-hSinglePass-stdDH-sha256kdf-scheme”  
“id-dhSinglePass-stdDH-sha384kdf-scheme”  
“id-dhSinglePass-stdDH-sha512kdf-scheme”

dhSinglePass-cofactorDH-sha1kdf-scheme OBJECT IDENTIFIER ::= { iso(1)  
identified-organization(3) tc68(133) country(16) x9(840) x9-63(63) chemes(0) 3 }

dhSinglePass-cofactorDH-sha224kdf-scheme OBJECT IDENTIFIER ::= {  
iso(1) identified-organization(3) certicom(132) schemes(1) 14 0 }

dhSinglePass-cofactorDH-sha256kdf-scheme OBJECT IDENTIFIER ::= {  
iso(1) identified-organization(3) certicom(132) schemes(1) 14 1 }

dhSinglePass-cofactorDH-sha384kdf-scheme OBJECT IDENTIFIER ::= {  
iso(1) identified-organization(3) certicom(132) schemes(1) 14 2 }

dhSinglePass-cofactorDH-sha512kdf-scheme OBJECT IDENTIFIER ::= {  
iso(1) identified-organization(3) certicom(132) schemes(1) 14 3 }

dhSinglePass-stdDH-sha1kdf-scheme OBJECT IDENTIFIER ::= { iso(1)  
identified-organization(3) tc68(133) country(16) x9(840) x9-63(63) chemes(0) 2 }

dhSinglePass-stdDH-sha224kdf-scheme OBJECT IDENTIFIER ::= { iso(1)  
identified-organization(3) certicom(132) schemes(1) 11 0 }

dhSinglePass-stdDH-sha256kdf-scheme OBJECT IDENTIFIER ::= { iso(1)  
identified-organization(3) certicom(132) schemes(1) 11 1 }

dhSinglePass-stdDH-sha384kdf-scheme OBJECT IDENTIFIER ::= { iso(1)  
identified-organization(3) certicom(132) schemes(1) 11 2 }

dhSinglePass-stdDH-sha512kdf-scheme OBJECT IDENTIFIER ::= { iso(1)  
identified-organization(3) certicom(132) schemes(1) 11 3 }

4.4.3.4.3. Алгоритми узгодження ключа, визначені ідентифікаторами у пункті 4.4.3.4.1 та 4.4.3.4.2, застосовується як для статичного, так і для динамічного механізму узгодження ключа; при цьому ознакою динамічного механізму є не нульове значення поля “originatorKey”, відповідно до пункту 4.4.2.2.2.

4.4.3.4.4. Алгоритми узгодження ключа, а саме ZZ-функція та KDF-функція, в групі точок еліптичної кривої (ECDH) визначені у розділі 5 цих Технічних специфікацій.

## V. Протокол узгодження ключа Діффі-Геллмана

### 5.1. Призначення та порядок виконання

5.1.1. Протокол узгодження ключа призначений для встановлення розділеної таємниці (створення узгоджувального ключа КШК) на основі використання асиметричних особистого ключа відправника та відкритого ключа одержувача і навпаки.

5.1.2. У Технічних специфікаціях визначено дві групи алгоритмів узгодження ключа Діффі-Геллмана: DH – в циклічній групі простого поля та ECDH – в групі точок еліптичної кривої.

5.1.3. Алгоритми узгодження ключа Діффі-Геллмана, що виконується відправником, містять такі кроки виконання:

- 1) Отримати параметри ключа відправника та ключа одержувача (із відкритих ключів їх сертифікатів X.509).
- 2) Порівняти параметри ключа відправника з параметрами ключа одержувача.
- 3) У разі еквівалентності параметрів встановлюється статичний механізм узгодження ключа, та перехід до кроку 5.
- 4) У разі не еквівалентності параметрів встановлюється динамічний механізм узгодження ключа, та відправник виконує генерацію ключової пари, використовуючи алгоритм та відповідні параметри ключа одержувача.
- 5) Виконати генерацію спільного секрету - ZZ-функцію генерації спільного секрету (ZZ) для визначеного протоколу, а саме: в циклічній групі простого поля (DH) або в групі точок еліптичної кривої (ECDH).
- 6) Виконати генерацію ключа шифрування ключа КШК - KDF-функцію (Key Derivation Function), функцію формування, на основі спільного секрету та додаткової інформації, ключа шифрування ключа КШК (KM - Keying Material) для заданого відправником алгоритму KeyWrapAlgorithm та заданого відправником алгоритму узгодження ключа через відповідний ідентифікатор, як визначено у пункті 4.4.3.3.

5.1.4. Алгоритми узгодження ключа Діффі-Геллмана, що виконується одержувачем, містять такі кроки виконання:

- 1) Завантажити сертифікат та особистий ключ одержувача та отримати параметри ключової пари відправника.
- 2) Отримати ASN.1 “EnvelopedData”
- 3) На основі аналізу “EnvelopedData” визначити механізм узгодження ключа. Ознакою динамічного механізму є ненульове значення поля “originatorKey”, відповідно до пункту 4.4.2.2.2.

- 4) Якщо механізм статичний, отримати сертифікат відправника. Сертифікат відправника може бути присутній у структурі “OriginatorInfo”, інакше повинен бути отриманий одержувачем із його сховища сертифікатів за даними з поля “OriginatorIdentifierOrKey”.
- 5) Виконати такі дії:
  - a. Отримати із сертифіката відправника відкритий ключ.
  - b. Отримати параметри відкритого ключа відправника.
  - c. Порівняти параметри ключа пари одержувача з параметрами відкритого ключа відправника.
  - d. У разі не еквівалентності параметрів – помилка, та припинити оброблення.
  - e. У разі еквівалентності параметрів перейти до кроку 6.
- 6) Якщо механізм динамічний отримати динамічний відкритий ключ відправника із структури “EnvelopedData”.
- 7) Виконати генерацію спільного секрету - ZZ-функцію генерації спільного секрету (ZZ) для визначеного протоколу, а саме: в циклічній групі простого поля (DH) або в групі точок еліптичної кривої (ECDH).
- 8) Виконати генерацію ключа шифрування ключа КШК - KDF-функцію (Key Derivation Function), функцію формування, на основі спільного секрету та додаткової інформації, ключа шифрування ключа КШК (КМ - Keying Material) для заданого алгоритму “KeyWrapAlgorithm” (який задано у структурі “EnvelopedData”) та заданого алгоритму узгодження ключа через відповідний ідентифікатор (який задано у структурі “EnvelopedData”), як визначено у пункті 4.4.3.3.

## 5.2. Параметри алгоритму узгодження ключа

Параметри алгоритму узгодження ключа називають “доменними параметрами” (“Domain Parameters”). У цій Специфікації доменні параметри не є об’єктом ASN.1 структури “захищені дані” (“EnvelopedData”), а використовуються виключно у внутрішніх процедурах - для визначення типу механізму (динамічний/ статичний) узгодження ключів та обчислення спільного секрету (ZZ-функція). Тому наведені у цьому пункті ASN.1 структури мають рекомендаційний характер.

### 5.2.1. Параметри алгоритму узгодження ключа у циклічній групі простого поля (DH)

5.2.1.1. Базуючись на міжнародних рекомендаціях RFC 3370 та Технічних специфікаціях “Формати представлення базових об’єктів”, доменні параметри алгоритму id-ESDH-ua можуть визначатися як ASN.1 структура:

```
DHParameters ::= SEQUENCE {
```

```

    p      INTEGER,
    q      INTEGER,
    a      INTEGER,
    validationParms GOST34310ValidationParms OPTIONAL, -- параметри
валідації
    dke    OCTET STRING OPTIONAL
}

GOST34310ValidationParms ::= SEQUENCE {
    x0     INTEGER,
    c      INTEGER,
    d      INTEGER OPTIONAL
}

```

5.2.1.2. Значення полів структури “DHParameters” наведено у таблиці 1.

Таблиця 1.

p	характеристика основного поля, просте число (modulus)
q	порядок циклічної підгрупи, фактор (factor, order of cyclic group)
a	твірний елемент циклічної підгрупи, генератор (generator)
dke	довгостроковий ключовий елемент (ДКЕ)
x0	початковий стан, що використовувався для генерації p, q (seed)
c	параметр датчика, що використовувався для генерації p, q.
d	довільне число, що використовувалося для генерації a, $1 < d < p - 1$

### 5.2.1.3. Операція порівняння доменних параметрів “DHParameters”

При визначенні механізму узгодження ключів, повинна виконуватися операція порівняння доменних параметрів. Якщо доменні параметри еквівалентні, то застосовується статичний механізм узгодження ключів, інакше – динамічний.

При виконанні операції порівняння параметрів повинні порівнюватися параметри p, a, q та додатково dke (у разі використання алгоритму “id-ESDH-ua”). Параметри валідації “GOST34310ValidationParms”, як не обов’язкові, не повинні використовуватися в операції порівняння.

## 5.2.2. Параметри алгоритму узгодження ключа в групі точок еліптичної кривої (ECDH)

5.2.2.1. Базуючись на міжнародних рекомендаціях RFC 5008, RFC 5480 та Технічних специфікаціях “Формати представлення базових об’єктів”, параметри алгоритму в групі точок еліптичної кривої (ECDH) можуть бути визначені як така ASN.1 структура:

```

ECDHParameters ::= SEQUENCE {

```

```

q    INTEGER,
FR   INTEGER,
a    INTEGER,
b    INTEGER,
G    ECPPoint,
n    INTEGER,
h    INTEGER,
dke  OCTET STRING OPTIONAL
}

```

де

$q$  – довжина поля (field size) в бітах, що рівна степені основного поля ( $m$ );  
 $FR$  – індикатор представлення поля або зведений поліном (reduction polynomial) (алгоритм обчислення наведено у пункті 5.4.2.2);

$a$  та  $b$  - два елементи поля, які визначають криву (коефіцієнти рівняння еліптичної кривої);

$G$  – базова точка еліптичної кривої (Base Point) з координатами ( $x_G, y_G$ );

$n$  - порядок базової точки (order of the point)  $G$ ;

$h$  - кофактор (що еквівалентний порядку кривої поділеному на  $n$ ); для еліптичних кривих з ДСТУ 4145-2002  $h = 2$ .

#### 5.2.2.2. Зображення точки еліптичної кривої ECPPoint

Значенням ECPPoint повинен бути рядок байтів, який представляє закодовану точку еліптичної кривої:

ECPPoint ::= OCTET STRING

#### 5.2.2.3. Процедура кодування точки (Point-to-Octet-String Conversion)

Вхідні дані:

Точка еліптичної кривої  $P = (X_p, Y_p)$ , яка не є нульовою,

Вихідні дані:

Рядок байтів PO – зображення у не стисненому форматі (uncompressed form) точки  $P$  як рядка байтів.

Процедура:

1. Байт PC встановлюється рівним 0x04 (ознака не стисненого формату):

$PC = 0x04$ .

2. Результуючий рядок байтів PO повинен бути об'єднанням (конкатенацією):

$PO = PC \parallel X_p \parallel Y_p$ .

Рядок байтів для представлення нульового елемента групи точок еліптичної кривої  $O = (0, 0)$  (infinity) повинен бути один нульовий байт:

PO = 0x00.

#### 5.2.2.4. Процедура обчислення FR

Поліномом є примітивний многочлен, наведений у таблиці 1 ДСТУ 4145-2002.

Значенням зведеного поліному є ціле число, як рядок біт.

Для оптимального нормального базису  $FR = 0$ .

Для поліноміального базису значення FR обчислюється (визначається) таким чином:

Нехай

$m$  - степень основного поля,

$ks[len]$  – масив цілих чисел  $ks[0]=k3$ ,  $ks[1]=k2$ ,  $ks[2]=k1$ , що є степенями примітивного многочлена

$$x^m + x^{k3} + x^{k2} + x^{k1} + 1,$$

де

$$m > k3 > k2 > k1 \geq 1$$

$len$  – довжина масиву  $ks$ , для тричлена (trinomial)  $len = 1$ , та для п'ятичлена (pentanomial)  $len = 3$ ; якщо  $len = 1$ , то  $k2 = k1 = 0$ .

Алгоритм визначення FR як рядка бітів:

1. Встановити  $FR = 1$  (встановити біт 0);
2. Встановити у FR біт  $m$ , та відповідно біти  $k1$ ,  $k2$ ,  $k3$ .

Отримаємо FR як рядок бітів.

#### 5.2.2.5. Операція порівняння доменних параметрів “ECDHParameters”

При визначенні механізму узгодження ключів, повинна виконуватися операція порівняння доменних параметрів. Якщо доменні параметри еквівалентні, то застосовується статичний механізм узгодження ключів, інакше – динамічний.

Порівнююватися повинні параметри структури ECDHParameters, яка наведена у пункті 5.4.2.1. Порівняння повинно виконуватися по-компонентно (еквівалентність параметрів  $q$ ,  $FR$  і т.д.) або як порівняння масивів байт DER-кодованої структури ECDHParameters.

#### 5.3. Обчислення спільного секрету

Обчислення спільного секрету не залежить від механізму узгодження ключів (статичний чи динамічний).

Обчислення спільного секрету відправником виконується на основі:

- особистого ключа відправника, та



- відкритого ключа одержувача.

Обчислення спільного секрету одержувачем виконується на основі:

- особистого ключа одержувача, та

- відкритого ключа відправника.

Процедура обчислення спільного секрету у цій Технічній специфікації називається ZZ-функція. Результатом обчислення спільного секрету є ZZ спільний секрет, як елемент поля, в якому виконується обчислення.

### 5.3.1. Обчислення спільного секрету у циклічній групі простого поля (DH)

5.3.1.1. Обчислення спільного секрету у циклічній групі простого поля (DH) ґрунтується на міжнародних рекомендаціях RFC 2631 та національному стандарті ДСТУ ISO/IEC 11770-3:2002 та виконується наступним чином:

$$Z = (y_b \wedge x_a) \bmod p = (y_a \wedge x_b) \bmod p,$$

де

Z – спільний секрет, як елемент поля, в якому виконується обчислення,

“ $\wedge$ ” – означає операцію піднесення до степеню,

“ $x_a$ ” – особистий ключ відправника,

“ $y_a$ ” – відкритий ключ відправника,

“ $x_b$ ” – особистий ключ одержувача,

“ $y_b$ ” – відкритий ключ одержувача,

“ $p$ ” – характеристика поля, просте число (odd prime)

“ $a$ ” – твірний елемент циклічної підгрупи, генератор (generator).

Отже, відправником виконується обчислення за формулою:

$$Z = (y_b \wedge x_a) \bmod p.$$

Одержувачем виконується обчислення за формулою:

$$Z = (y_a \wedge x_b) \bmod p.$$

### 5.3.1.2. Алгоритм DH не повинен застосовуватися якщо:

$$a^x = a \pmod{p} \text{ або } a^y = a \pmod{p}.$$

де

$x$  - особистий ключ,

$y$  - відкритий ключ.

### 5.3.2. Обчислення спільного секрету в групі точок еліптичної кривої (ECDH)

5.3.2.1. Обчислення спільного секрету з *кофакторним множенням* в групі точок еліптичної кривої (ECDH) ґрунтується на національному стандарті ДСТУ ISO/IEC 15946-3:2006 та виконується наступним чином:

Відправником обчислюється значення точки  $K$  еліптичної кривої з координатами  $(x_k, y_k)$ :

$$K = d_a * h * P_b, \quad (1)$$

а отримувачем –

$$K = d_b * h * P_a, \quad (2)$$

де

$d_a$  – особистий ключ відправника,

$P_a$  – відкритий ключ відправника,

$d_b$  – особистий ключ одержувача,

$P_b$  – відкритий ключ одержувача,

$h$  – кофактор (для еліптичних кривих з ДСТУ 4145-2002  $h = 2$ ).

Спільним секретом є координата  $x_k$  точки  $K$ :

$$Z = x_k,$$

де

$Z$  – спільний секрет, як елемент поля, в якому виконується обчислення,

5.3.2.2. У разі обчислення спільного секрету *без кофакторного множення* значення кофактору у формулах (1) та (2) приймають рівним одиниці ( $h=1$ ).

5.3.2.3. Виконання операцій над точками еліптичної кривої, зображення даних, перевірка правильності загальних параметрів алгоритму та правильності ключів здійснюється згідно з національним стандартом України ДСТУ 4145-2002.

5.3.3. Перетворення елемента поля  $Z$  на рядок байтів  $ZZ$

Для використання у функціях формування ключа (KDF-функціях) спільного секрету  $Z$ , отриманого у пунктах 5.5.1 та 5.5.2, необхідно перетворити елемент поля  $Z$  на рядок байтів  $ZZ$  (Field-Element-to-Octet-String Conversion).

Таке перетворення повинно виконуватися наступним чином:

Нехай  $Z$  є елементом поля  $F_q$  (FFC DH) чи поля  $F_{2^m}$  (ECC DH). Результатом перетворення є рядок байтів  $ZZ$  довжини  $L$ .

1. Якщо  $Z$  є елементом поля  $F_q$ , то воно є додатнім цілим числом, тобто двійковим (бітовим) рядком (bit string), якщо  $Z$  є елементом поля  $F_{2^m}$ , то виконати перетворення елемента поля  $Z$  на додатне ціле число, як визначено у пункті 5.8 ДСТУ 4145-2002. Позначимо ціле число від  $Z$  як  $ZI$ .
2. Виконати перетворення цілого  $ZI$  на рядок байтів  $ZZ$  у форматі Big-Endian. Перетворення цілого  $ZI$  на рядок байтів  $ZZ$  у форматі Little-Endian наведено у пункті 1.3.14.2 Технічних специфікацій “Формати представлення базових об’єктів”. Формат Big-Endian має зворотній порядок байтів щодо формату Little-Endian.

При прямому розміщенні байт (Big-Endian) старший байт (big-end) повинен зберігатися за найменшою адресою (як байт з найменшим індексом байт-масиву), а при зворотньому розміщенні (Little-Endian) — за найбільшою, тобто за найменшою адресою повинен розмішуватися молодший байт (little-end).

### 5.3.3. Приклади перетворення елемента поля на рядок байтів у форматі Big-Endian

Приклад 1 (пункт 1.3.14.2 Технічних специфікацій “Формати представлення базових об’єктів”).

$Z = 1\ 1110\ 0011\ 0111$

$ZI = 7735$

$ZZ = 1E\ 37$

Примітка. У форматі Little-Endian кодується як послідовність байт “37 1E”.

Приклад 2.

$ZI = 65048$

$ZZ = 00\ FE\ 18$

Приклад 3.

$ZI = 1450621147818416422260325141022141529263703331843$

$ZB = 00\ FE\ 18\ 1A\ 1B\ 1C\ 17\ 19\ 16\ 10\ 13\ 0F\ 17\ 08\ 1B\ 02\ 18\ 16\ 14\ 14\ 03$

### 5.3.3. Приклади обчислення спільного секрету $ZZ$

#### 5.3.3.1. У циклічній групі простого поля (DH)

Приклад 1. FFC DH, ГОСТ 34.310, 512 біт

$p = C8\ 21\ B6\ 11\ FA\ FF\ 21\ 32\ 1B\ 4D\ 14\ 20\ D1\ 99\ 50\ 65\ 19\ C7\ 86\ 67\ B0\ 1E\ E5\ F0\ 4D\ DB\ BC\ A6\ C4\ BD\ AD\ C1\ 0A\ 83\ 11\ 98\ AA\ 98\ A9\ 2A\ 4C\ 77\ 80\ 2D\ C6\ 7D\ 24\ B6\ 1E\ 48\ F1\ 9D\ 3D\ E8\ A5\ 02\ 51\ A7\ 77\ 23\ 30\ 71\ FE\ BF$

$q = D8\ B3\ E2\ A9\ DF\ 80\ 7A\ 6A\ 57\ 8C\ CB\ 78\ 8D\ 68\ C1\ C1\ 9D\ FF\ B2\ A6\ 75\ 7C\ 6B\ EB\ CD\ 24\ 32\ EA\ 1E\ 78\ 71\ CD$

a=B0 A8 AF CE A2 6B 3B 43 4D 07 8E 80 0A 3D 53 9D 24 00 B8 63 6D BE  
55 D9 AF C9 07 A7 D3 76 58 75 65 7A A6 00 42 0F 64 EB 7E 51 F4 28 83 A2 26 94  
5B 73 7D EA 33 FA 84 E6 CF BB 95 2B 10 A0 BC 5D

xa=00 A5 B9 B3 86 56 13 64 51 29 27 2B 0D 42 99 5B 9E E3 73 49 5A E2 6A  
84 12 B2 C1 E0 48 04 F7 B7 E3

ya=0A 6B 36 4E 56 8D EC B7 75 53 4B 99 82 53 C9 38 CE 45 E5 90 61 BB  
DC 53 A9 67 B8 F9 34 C5 D7 DA AE 74 63 A1 4E D5 91 A5 D6 91 AE 65 47 BC  
EA 3E D3 FF 2B 81 D4 DD 1D 0D 7F 73 10 46 C0 13 4E 3A

xb=45 A7 63 84 56 AE F3 7C 1A 47 AB F1 5D 71 11 44 2F BC 9F C2 9B 87  
AD B8 F6 54 FF EE 9C C8 72 A5

yb=53 2C 26 ED 6D 50 33 97 7B A6 44 17 59 5B 20 5A 85 5B F0 D6 2F BE  
20 42 F2 C2 53 2E 4A 14 0C 05 0E 04 A4 32 9D EC A0 70 35 21 79 F7 9E 95 22 CB  
38 E6 2D 0C 06 E6 00 0C 8E 8B 22 B3 91 50 40 A3

ZZ=75 DF 2D 1F 03 12 C2 B7 F2 2B 8D 55 41 21 E9 84 A9 98 C5 59 2E F6  
A9 CB E8 79 84 82 F1 AD D6 65 F2 94 8D CC 74 FF C2 CC EC D0 B9 DA 70 97  
BE 8B 9D D6 B0 6C BB BA 64 89 AC 91 DC 85 67 9E 2E 0E

Приклад 2. FFC DH, ГОСТ 34.310, 1024 бит

p=E4 C6 F8 34 11 B1 55 9B 99 5D 5E 15 30 84 50 98 C3 0D CB 3E 2A A1 C2  
BD BE ED 4B EF 7D 92 2F DF 04 E1 A7 55 08 8F 46 39 85 19 20 51 DE 7A 06 03  
0D B6 36 2F 5F C3 2A 99 88 02 96 27 66 7C BC B1 26 9A A2 11 11 56 3D A2 47 13  
31 A2 88 9F 35 C7 52 CB E6 FF 02 25 61 43 DB 9A CA 45 87 C9 3E B6 F9 D0 51  
78 54 7F F8 43 9C FA AB E2 37 9A 7D 9E 14 C5 EA 84 10 17 BB CA CA 9C 35 9C  
A6 B3 8A 6F

q=D0 BD 51 FB 45 F3 E4 A6 C8 16 97 D4 63 16 3B 03 1D F0 46 DF 19 05 4F  
BF D2 50 56 87 86 71 BF 91

a=BD A6 75 95 BC 18 A3 E5 27 13 A9 D4 E1 08 6E C3 E0 99 08 50 70 B0 28  
57 59 57 E4 5B 28 DD 72 D8 2D 41 D2 B7 93 06 91 B5 BC 6C 79 81 86 39 6F 53 48  
53 97 F0 F2 70 73 56 3A 79 56 0D 93 76 00 9C 9F 4B 63 CA 6C 9E E0 7D 12 B1 85  
62 A8 CE 19 2E F9 1C 33 2F C4 1A AF 8A 59 E6 97 E8 D9 AE 0F 7E E5 07 D7 B4  
3F 60 F4 A6 D0 8E 71 BE 08 8A E3 82 8B EC 91 BC 3D C6 B8 19 97 5D B3 E2 2D  
98 AB A8

xa=00 93 49 C5 03 C3 09 FC 95 B0 6E 81 CB F5 0C 7E 8D 54 E4 1B 1A 3B  
F8 70 43 2A FC 14 A6 FA 80 A7 D5

ya=76 39 EF E2 1B 30 8D C4 6B F3 3F C6 9B AB 2F 12 EF 2E D5 18 E9 89  
BB 3A D1 09 4C 9F F7 27 0C D2 C4 01 9A D8 2B 46 63 EA 77 24 E7 0F EF E2 A8  
B1 B1 12 9E 2F 2A B0 A7 A5 50 B8 F1 A7 D4 06 07 E2 EE 95 52 3A F1 6C 07 DC  
C5 57 24 FD E7 9D EC 72 66 FD 6C DE 70 6C D4 BA C1 70 E3 C6 D8 56 01 12 E8  
9F C3 2C A5 0F D2 74 1F 59 CF 41 98 CD 17 CC 88 DF 42 24 81 3A 5E E0 93 00  
B8 2C 91 E2 B2 BD

xb=30 18 1A 92 B5 E0 54 42 97 E5 10 AF 20 51 FB C8 56 26 20 97 AD A7 47  
5A 5B 70 15 67 5E 08 9D F9

yb=00 C0 6B DD D0 A4 0F CD 55 BA 79 54 A3 E7 9C DF F9 24 0C C0 48 B4  
EA FA A5 91 AA 7E 75 6E 57 27 A1 98 4F 4D BC 01 3A C7 7A B8 16 A6 7C EA  
53 75 8C 7E 2D A8 8F C1 25 30 C8 73 C6 CD F2 DD 1A FD 27 0F 14 7F 1F 49 BD

9B E7 68 04 3E B8 FC 95 A4 3A 0D 68 72 6B 8A C2 96 CD D1 05 88 89 CE 4D CA  
B9 BD CE 6B 3A E2 D7 96 34 FC AA 07 85 8B A4 3F 54 B4 CD E8 01 36 DD 1E  
83 49 DE AE 0F 5E CE 8E DA

ZZ=BD 7F 88 9E 48 B7 D6 2A 37 1A 6C 52 B9 2F 90 24 A2 D6 B5 05 82 04  
5E 31 E2 94 95 99 01 54 7D BF 6D 90 50 B0 A0 AC D0 50 FE 96 9A AE EF B8 51  
85 15 4D 9E 1D F7 D7 F4 86 DB 00 13 31 86 C6 B7 F8 4D 66 93 F4 F0 83 C6 3C  
A7 79 B9 02 C5 B8 9D 7F 74 16 CD AF CA 69 55 55 61 C5 F4 2F 53 B4 89 B4 7A  
A0 F5 31 21 91 42 1F 57 01 C6 60 54 3B B7 22 2D F6 58 6A E8 61 74 75 E7 52 86  
6B 84 51 5D BF

### 5.3.3.2. В групі точок еліптичної кривої (ECDH)

Приклад 1. ECC DH, ДСТУ4145 ПБ, m=163

curveOID=1.2.804.2.1.1.1.3.1.1.2.0

dA=00 03 04 99 1F 9A C1 A8 09 4F 6F BF A0 09 25 0D 4A 80 99 32 0D 55

QA<sub>x</sub>=30 01 C5 6A 32 99 13 07 62 6D 09 B5 FF 06 9C 6C B8 0B 79 4D 39 BD

QA<sub>y</sub>=00 01 B9 85 8C 28 B9 BC DC A1 7B A3 5E 6D 5F 03 4B 23 AA 74 33

C7

dB=00 01 FC 86 17 11 60 74 A8 FF 81 B4 2F 85 CA 25 16 CF 11 CE 2E 13

QB<sub>x</sub>=00 01 F3 36 B1 E8 02 4B E4 AB E9 2D 26 CA 7F 30 22 0E 16 50 BC 1A

QB<sub>y</sub>=00 02 F8 75 7B 1E 7E 72 E3 E5 46 B2 60 41 77 46 3D 3F C3 BE 63 C9

ZZ=07 F8 4A DB A6 24 57 C5 DA 5D 95 94 47 C4 F6 C1 86 4C 9B 28 8E

Приклад 2. ECC DH, ДСТУ4145 ПБ, m=431

curveOID=1.2.804.2.1.1.1.3.1.1.2.9

dA=4B 3A 17 07 F8 70 D0 C1 D4 CE 43 8E 88 AA 2B 36 15 06 91 62 86 F3  
6F F3 5F 9C BE 9C 0F BD BE 1F 77 6B B5 C2 58 78 BA E1 C5 49 58 D1 B0 39 B1  
2F F5 47 E0 08 63

QA<sub>x</sub>=12 21 63 2E 63 D9 0A EB 4D B4 31 B9 9D E5 A8 7B 74 18 02 3D D7  
12 B5 01 15 64 14 A0 F8 4C 07 41 B1 27 87 65 E6 3E A7 14 B3 D4 B6 8E A6 A1 04  
53 08 B6 79 AD 96 69

QA<sub>y</sub>=40 3D 64 C6 F5 15 62 B7 00 AC CD 27 AB CA 66 2F 87 97 74 4E 11 21  
68 01 5E 0A 14 3A D0 29 62 7A 13 78 EB BA 93 48 70 D1 64 12 16 A4 8A 3F 10 17  
FA 22 11 49 F6 83

dB=06 FE 21 1C 55 5C B9 D3 A2 7B 7E C2 E2 FF BB 4C B1 67 EE 86 BB 7F  
11 1E 7B CD 8F 65 64 E8 83 5B 09 E0 47 B9 28 6F BA 7F 83 50 78 79 BF EE 4C 33  
EC BA 41 C5 DA B5

QB<sub>x</sub>=50 B0 73 73 CF EE 5B 3F 7C 28 70 99 E3 B3 06 AE B1 69 0C 7E 66 92  
6D C5 02 D1 88 D8 81 FC 17 DC 75 AF D6 CE 2B 1E 9C ED 7C 16 FF D3 29 CF 41  
1A 4B 0A BC 98 53 F5

QB<sub>y</sub>=56 9F F5 92 E5 8A 9A 4F C8 EF 7E B9 97 A1 AA AC 46 10 5D 90 32  
F7 89 D7 7C 6B 3F 7C 7B AE D2 BE F4 C2 AE 10 42 E2 B2 5A 98 AD 00 74 9A 43  
63 91 CC 98 F2 7C 43 5A

ZZ=4B CA 28 D6 48 A5 0D EE F8 5F 8A 34 73 5B AE 5C 1A FE A7 2D 35  
15 E1 66 39 E5 F1 66 DD 0A 47 ED E3 33 EA 5A B3 41 5D BC 4F DB B7 B6 8B E2  
49 FF 3C 5B 75 F8 2B 55

Приклад 3. ECC DH, ДСТУ4145 ОНБ, m=173

curveOID=1.2.804.2.1.1.1.3.1.2.2.0

dA=02 4D D6 E9 66 40 D1 67 F3 41 6B 77 C6 21 1B 86 20 0C 10 CF A6 60

9C

QA<sub>x</sub>=12 BC 80 94 27 D6 11 D7 AE 2F 5A 2C EE 2C A5 8A D9 C3 CF 27 1D

7F

QA<sub>y</sub>=15 D4 28 1B B8 12 E5 A5 C2 16 BE DB B6 70 30 3C 34 0B 0A E4 CD

dB=02 B1 63 9B 21 01 3E 34 E6 0A 3F 98 D3 04 A4 57 12 EB 10 18 EF 6B

QB<sub>x</sub>=D3 C3 C6 32 76 EE 77 38 BA E5 D0 6D 01 F3 F5 2D B1 D2 F8 C3 CF

14

QB<sub>y</sub>=06 CB 9C 6D BC 22 A2 D0 83 C0 A4 38 27 83 F5 29 6F D3 CF 7F 3B

ZZ=2C E3 D0 85 F5 93 BF 26 64 25 35 BB 16 A7 F7 DD 53 10 46 A7 3D 2A

#### 5.4. Функція формування ключа КШК (KDF-функція)

##### 5.4.1. Призначення та кроки виконання

KDF-функція призначена для формування ключа шифрування ключа (КШК) на основі спільного секрету ZZ та іншої інформації.

Кроки виконання:

1. На основі спільного секрету ZZ та іншої інформації сформувати ключовий матеріал (КМ).

2. На основі ключового матеріалу створити ключ шифрування ключа КШК для заданого алгоритму шифрування.

##### 5.4.1. KDF-функція у циклічній групі простого поля (DH)

Функція формування ключа (KDF-функція) у циклічній групі простого поля (DHKDF) ґрунтується на міжнародних рекомендаціях RFC 2631 та національному стандарті ДСТУ ISO/IEC 15946-3:2002 (Додаток А.2. Функція формування ключа ANSI X9.42).

5.4.1.1. Формування ключового матеріалу КМ виконується відповідно до алгоритму:

$$\text{КМ} = \text{H} (\text{ZZ} \parallel \text{OtherInfo})$$

де

КМ – ключовий матеріал (рядок байтів), довжина якого залежить від геш алгоритму H;

H – геш-функція, визначається ідентифікатором алгоритму узгодження, що визначені у пункті 4.4.3.3;

ZZ – спільний секрет, обчислений відповідно до пункту 5.3;

OtherInfo – DER-кодована ASN.1 структура;  
|| – означає операцію конкатенації.

#### 5.4.1.2. Структура “OtherInfo”

“OtherInfo” має таку ASN.1 структуру:

```
OtherInfo ::= SEQUENCE {  
    keyInfo      KeySpecificInfo,  
    partyAInfo   [0] OCTET STRING OPTIONAL,  
    suppPubInfo  [2] OCTET STRING  
}  
  
KeySpecificInfo ::= SEQUENCE {  
    algorithm     OBJECT IDENTIFIER,  
    counter       OCTET STRING SIZE (4..4) }
```

Зазначені ASN.1 структури використовують EXPLICIT (“явний”) теги (В ASN.1, EXPLICIT теги означає, що “не явний” (implicit) використовується тільки тоді, коли явно визначено IMPLICIT).

#### 5.4.1.3. Поля структури “OtherInfo”

“algorithm” – є об’єктним ідентифікатором (ASN.1 OID) алгоритму KeyWrapAlgorithm - ключа шифрування ключа, на якому повинен бути зашифрований ключ шифрування повідомлення (даних). Звертаємо увагу, що це не є ідентифікатор алгоритму (AlgorithmIdentifier), а лише його об’єктний ідентифікатор, - параметри алгоритму тут не використовуються.

“counter” – це 32-х бітне число, яке представлено як бітовий вектор (чотири байти), записаного у зворотному порядку. Початковим значенням “counter” є 1 (одиниця) для будь-якого ZZ, а його бітовий вектор є “00 00 00 01” (hex); значення counter збільшується на одиницю (incremented) з кожним циклом виконання функції формування ключового матеріалу КМ (пункт 5.4.1.4) під час генерації ключа КШК;

“partyAInfo” – це випадковий рядок, який генерує відправник. В CMS це значення розміщується в полі “ukm” (“UserKeyingMaterial”) (закодоване як OCTET STRING) структури “KeyAgreeRecipientInfo”. Довжина “partyAInfo” повинна бути 512 біт (64 байти);

“suppPubInfo” - це довжина сформованого ключа КШК в бітах, представлена як бітовий вектор (чотири байти) 32-бітного числа. Наприклад, ключ 192 біт повинен бути представлений як бітовий вектор “00 00 00 C0” (hex).

#### 5.4.1.4. Формування ключа КШК

Так як довжина ключового матеріалу (рядок байтів) КМ, сформованого відповідно до пункту 5.4.1.1, залежить від алгоритму Н, та може не бути рівною довжині ключа КШК (як рядка байтів), то використовується такий алгоритм:

А) Якщо довжина КМ більше, ніж довжина КШК, то за КШК приймають перші N байтів КМ, де N – довжина КШК.

Б) Якщо довжина КМ дорівнює довжині КШК, то за КШК приймають КМ.

В) Якщо довжина КМ менше, ніж довжина КШК, то:

Встановлюється значення counter = 1, та формується КМ1.

Встановлюється значення counter = 2, та виконується КМ2.

.....

(кількість кроків формування КМі визначається необхідною довжиною ключа КШК).

Отримані блоки ключового матеріалу об'єднуються (concatenation) для отримання необхідної довжини (останні байти останнього блоку КМі відкидаються):

$$KM = KM1 \parallel KM2 \parallel \dots$$

Таким чином, що довжина КМ повинна бути рівною довжині ключа КШК.

#### 5.4.1.5. Щодо опціональності параметру “partyAInfo”

Якщо параметр “partyAInfo”, як не обов'язковий, не буде використовуватися, то у випадках А та Б для різних повідомлень буде формуватися один і той же ключ шифрування КШК. Для того щоб уникнути цього, у разі статичного механізму вимагається (а у разі динамічного – рекомендується) генерувати випадкове значення “partyAInfo” для кожного повідомлення, та використовувати під час формування КШК.

#### 5.4.1.5. Приклади обчислення ключа КШК у циклічній групі простого поля (DH)

Примітка. В наведених прикладах у якості KeyWrapAlgorithm (“algorithm”) використовується ГОСТ 28147-87 у режимі гамування (“id-gost28147-ofb”, розділ 7) чи гамування із зворотнім зв'язком (“id-gost28147-cfb”, розділ 7). Ці алгоритми не є Wrap-алгоритмами (які викладені у розділ 6), і використовуються тут лише для цілей наведення прикладів.

Сформований ключ КШК позначається у прикладах через КЕК, його довжина в бітах – через keyLen.

Приклади наведені для алгоритму узгодження ключа “id-ESDH-ua”, тобто з використанням геш-функції ГОСТ 34.311. ДКЕ позначено через sBox (SBOX-1 – це ДКЕ №1 із переліку рекомендованих Інструкцією №114).



Приклад 1. ГОСТ 34.310 512 біт, “id-gost28147-cfb” KeyWrapAlgorithm

```
algorithm=1.2.804.2.1.1.1.1.1.3
sBox=SBOX-1
keyLen=256
partyAInfo=0123456789abcdeffedcba98765432010123456789abcdeffedcba987
65432010123456789abcdeffedcba98765432010123456789abcdeffedcba9876543201
p=C0 23 F2 3D C3 75 65 48 2E 1D 32 B8 A2 91 70 B7 42 41 B5 CB F4 11 53
5E 09 8D 36 36 F4 36 96 22 BF 2C 8A FC AC 2A AF 37 CD 30 6E F3 15 31 43 D9
CA D0 13 7F 63 1E D0 46 A0 5E FA 30 09 CC 8E D7
q=B6 1F 3C B5 CF 9E 10 E0 37 6A 6E F0 D3 5B 5F 1E CB 6E 6A 74 34 C2
D7 1F D7 AA C9 16 3F 05 AB 67
a=8C C2 DA 25 D2 91 22 56 E4 55 FE 42 1A 2D 5F 09 F2 76 2C 53 F6 B5 AA
2F C3 90 FC 7B 53 36 2E 3A 10 A6 70 7F E8 67 A3 4D 49 A0 95 F7 E6 93 5F 10 D7
7C FA D4 5A 23 3F 27 25 9D 1C 91 89 EB C5 F6
xa=2C B2 A7 A0 9E 78 D0 B6 BA 3F D5 18 2E 1E 1A F1 55 C5 16 6E F8 C7
D3 8D 2D F8 72 D4 72 03 45 64
ya=00 B5 0F 40 2E 25 9A 57 15 0E 41 1B 23 C8 D3 26 43 B3 31 3D BF 80 A4
C9 6A 76 51 F4 65 AA 27 DF 83 61 AF 53 6C 66 13 C9 CE CC BD EC 28 8A E7 78
FA 46 2F 87 CC D1 BA C6 67 61 C7 FD 90 31 44 11 94
xb=00 8C C9 E6 E3 FA 03 BE 30 96 BA 44 3B D7 2A BC 1C BB 1F 35 73 05
A2 EC 28 EA 73 31 F4 15 0B 19 A2
yb=7F 28 D7 A4 98 71 1F 5E 1D 8D D4 69 C7 9D 32 15 BA CC 26 22 D6 C4
C3 26 C0 44 B3 04 8B 77 17 8D 27 95 11 F0 9B 01 A6 B9 00 5C 64 54 9A 30 E8 F2
92 3E 6C 21 60 E1 10 E5 30 81 D5 3A CE CF D9 9D
Z=30 4B 47 CA 74 52 C5 F6 25 33 60 11 F2 C4 CB E7 96 0D EA 88 02 D0 3C
A6 2B 09 DA DE FD 28 09 61 76 B7 16 8E 9C 7D E3 3F 9D D3 9E 92 45 CA AD 17
95 2F F6 81 B2 F5 6D D4 51 16 74 B4 B0 92 23 F3
KEK=7B 06 44 1A 06 0E 50 C3 F7 C0 0E 45 0E 8F 0F 87 9C 2E 6F 85 F5 2E
C2 31 AA BE AE 9B 07 22 D8 F9
```

Приклад 2. ГОСТ 34.310 1024 біт, “id-gost28147-ofb”KeyWrapAlgorithm

```
algorithm=1.2.804.2.1.1.1.1.1.2
sBox=SBOX-3
keyLen=256
partyAInfo=0123456789abcdeffedcba98765432010123456789abcdeffedcba987
65432010123456789abcdeffedcba98765432010123456789abcdeffedcba9876543201
p=E4 C6 F8 34 11 B1 55 9B 99 5D 5E 15 30 84 50 98 C3 0D CB 3E 2A A1 C2
BD BE ED 4B EF 7D 92 2F DF 04 E1 A7 55 08 8F 46 39 85 19 20 51 DE 7A 06 03
0D B6 36 2F 5F C3 2A 99 88 02 96 27 66 7C BC B1 26 9A A2 11 11 56 3D A2 47 13
31 A2 88 9F 35 C7 52 CB E6 FF 02 25 61 43 DB 9A CA 45 87 C9 3E B6 F9 D0 51
78 54 7F F8 43 9C FA AB E2 37 9A 7D 9E 14 C5 EA 84 10 17 BB CA CA 9C 35 9C
A6 B3 8A 6F
```

q=D0 BD 51 FB 45 F3 E4 A6 C8 16 97 D4 63 16 3B 03 1D F0 46 DF 19 05 4F  
BF D2 50 56 87 86 71 BF 91

a=BD A6 75 95 BC 18 A3 E5 27 13 A9 D4 E1 08 6E C3 E0 99 08 50 70 B0 28  
57 59 57 E4 5B 28 DD 72 D8 2D 41 D2 B7 93 06 91 B5 BC 6C 79 81 86 39 6F 53 48  
53 97 F0 F2 70 73 56 3A 79 56 0D 93 76 00 9C 9F 4B 63 CA 6C 9E E0 7D 12 B1 85  
62 A8 CE 19 2E F9 1C 33 2F C4 1A AF 8A 59 E6 97 E8 D9 AE 0F 7E E5 07 D7 B4  
3F 60 F4 A6 D0 8E 71 BE 08 8A E3 82 8B EC 91 BC 3D C6 B8 19 97 5D B3 E2 2D  
98 AB A8

xa=00 93 49 C5 03 C3 09 FC 95 B0 6E 81 CB F5 0C 7E 8D 54 E4 1B 1A 3B  
F8 70 43 2A FC 14 A6 FA 80 A7 D5

ya=76 39 EF E2 1B 30 8D C4 6B F3 3F C6 9B AB 2F 12 EF 2E D5 18 E9 89  
BB 3A D1 09 4C 9F F7 27 0C D2 C4 01 9A D8 2B 46 63 EA 77 24 E7 0F EF E2 A8  
B1 B1 12 9E 2F 2A B0 A7 A5 50 B8 F1 A7 D4 06 07 E2 EE 95 52 3A F1 6C 07 DC  
C5 57 24 FD E7 9D EC 72 66 FD 6C DE 70 6C D4 BA C1 70 E3 C6 D8 56 01 12 E8  
9F C3 2C A5 0F D2 74 1F 59 CF 41 98 CD 17 CC 88 DF 42 24 81 3A 5E E0 93 00  
B8 2C 91 E2 B2 BD

xb=03 84 40 38 A3 69 BA 43 15 BB 3B 64 27 15 9C CE AC 37 E7 63 07 B5  
B6 F5 23 EA 01 0A 0F 7A 04 BD

yb=00 D1 1F 56 A1 40 34 9A DC 48 F0 BD C5 5B BF B6 4E BC 59 5C 62 5A  
AB 93 EB E6 B6 49 C4 88 B2 E3 AB 51 76 58 BC 38 E7 EC 6A 2B A3 DF 03 2B 62  
64 0E C0 92 B7 1B 61 EB FD 17 A3 CD 68 75 B8 14 C9 51 B8 36 D2 0C 32 CD 7D  
6E 79 54 E7 06 4A 37 8E 77 88 2C 7A 09 BE 01 27 81 8C FF 88 53 9A C9 0E D3 FD  
BF 3E 76 65 13 D0 EA FA AF AB 17 01 58 92 70 7C 1D D5 60 8E 63 7D 4D 8E 5A  
3F 47 ED E3 FC 10

Z=27 1C 54 DE AA AD 82 99 C1 7A C2 95 DE 74 0A 04 60 57 CD D2 A8 3E  
A4 27 A6 D1 AF 53 CC 0C 71 E7 8E B7 7B 1F 41 D3 97 45 9F EF 39 83 C9 7E 3F  
53 74 BD 2B 82 F2 B8 E6 FF 0A 08 F2 E0 1E F5 4C BB EE DE FA 7E 23 D6 A4 66  
E1 DF 8E E6 D8 DB 6D C9 2F 7D 67 9A 9F 23 F5 BE 16 5A 26 D6 1D 7D B2 0C  
C8 8F F7 E0 C1 CA E0 E6 1C BF EE 06 AD 52 16 F8 68 AC D3 16 FF B0 30 E6 BF  
49 3E F8 9D 18 DA FD

KEK=36 C0 9A A8 83 A4 B5 83 EB C8 ED 28 17 7E 6C 35 BF A7 F4 57 8A  
25 8E 4E 5C AE 35 05 82 FB D2 A7

Приклад 3. ГОСТ 34.310 1024 бит, “id-gost28147-ofb”KeyWrapAlgorithm

algorithm=1.2.804.2.1.1.1.1.1.2

sBox=SBOX-4

keyLen=256

partyAInfo=

p=E4 C6 F8 34 11 B1 55 9B 99 5D 5E 15 30 84 50 98 C3 0D CB 3E 2A A1 C2  
BD BE ED 4B EF 7D 92 2F DF 04 E1 A7 55 08 8F 46 39 85 19 20 51 DE 7A 06 03  
0D B6 36 2F 5F C3 2A 99 88 02 96 27 66 7C BC B1 26 9A A2 11 11 56 3D A2 47 13  
31 A2 88 9F 35 C7 52 CB E6 FF 02 25 61 43 DB 9A CA 45 87 C9 3E B6 F9 D0 51  
78 54 7F F8 43 9C FA AB E2 37 9A 7D 9E 14 C5 EA 84 10 17 BB CA CA 9C 35 9C  
A6 B3 8A 6F

q=D0 BD 51 FB 45 F3 E4 A6 C8 16 97 D4 63 16 3B 03 1D F0 46 DF 19 05 4F  
BF D2 50 56 87 86 71 BF 91

a=BD A6 75 95 BC 18 A3 E5 27 13 A9 D4 E1 08 6E C3 E0 99 08 50 70 B0 28  
57 59 57 E4 5B 28 DD 72 D8 2D 41 D2 B7 93 06 91 B5 BC 6C 79 81 86 39 6F 53 48  
53 97 F0 F2 70 73 56 3A 79 56 0D 93 76 00 9C 9F 4B 63 CA 6C 9E E0 7D 12 B1 85  
62 A8 CE 19 2E F9 1C 33 2F C4 1A AF 8A 59 E6 97 E8 D9 AE 0F 7E E5 07 D7 B4  
3F 60 F4 A6 D0 8E 71 BE 08 8A E3 82 8B EC 91 BC 3D C6 B8 19 97 5D B3 E2 2D  
98 AB A8

xa=00 93 49 C5 03 C3 09 FC 95 B0 6E 81 CB F5 0C 7E 8D 54 E4 1B 1A 3B  
F8 70 43 2A FC 14 A6 FA 80 A7 D5

ya=76 39 EF E2 1B 30 8D C4 6B F3 3F C6 9B AB 2F 12 EF 2E D5 18 E9 89  
BB 3A D1 09 4C 9F F7 27 0C D2 C4 01 9A D8 2B 46 63 EA 77 24 E7 0F EF E2 A8  
B1 B1 12 9E 2F 2A B0 A7 A5 50 B8 F1 A7 D4 06 07 E2 EE 95 52 3A F1 6C 07 DC  
C5 57 24 FD E7 9D EC 72 66 FD 6C DE 70 6C D4 BA C1 70 E3 C6 D8 56 01 12 E8  
9F C3 2C A5 0F D2 74 1F 59 CF 41 98 CD 17 CC 88 DF 42 24 81 3A 5E E0 93 00  
B8 2C 91 E2 B2 BD

xb=03 84 40 38 A3 69 BA 43 15 BB 3B 64 27 15 9C CE AC 37 E7 63 07 B5  
B6 F5 23 EA 01 0A 0F 7A 04 BD

yb=00 D1 1F 56 A1 40 34 9A DC 48 F0 BD C5 5B BF B6 4E BC 59 5C 62 5A  
AB 93 EB E6 B6 49 C4 88 B2 E3 AB 51 76 58 BC 38 E7 EC 6A 2B A3 DF 03 2B 62  
64 0E C0 92 B7 1B 61 EB FD 17 A3 CD 68 75 B8 14 C9 51 B8 36 D2 0C 32 CD 7D  
6E 79 54 E7 06 4A 37 8E 77 88 2C 7A 09 BE 01 27 81 8C FF 88 53 9A C9 0E D3 FD  
BF 3E 76 65 13 D0 EA FA AF AB 17 01 58 92 70 7C 1D D5 60 8E 63 7D 4D 8E 5A  
3F 47 ED E3 FC 10

Z=27 1C 54 DE AA AD 82 99 C1 7A C2 95 DE 74 0A 04 60 57 CD D2 A8 3E  
A4 27 A6 D1 AF 53 CC 0C 71 E7 8E B7 7B 1F 41 D3 97 45 9F EF 39 83 C9 7E 3F  
53 74 BD 2B 82 F2 B8 E6 FF 0A 08 F2 E0 1E F5 4C BB EE DE FA 7E 23 D6 A4 66  
E1 DF 8E E6 D8 DB 6D C9 2F 7D 67 9A 9F 23 F5 BE 16 5A 26 D6 1D 7D B2 0C  
C8 8F F7 E0 C1 CA E0 E6 1C BF EE 06 AD 52 16 F8 68 AC D3 16 FF B0 30 E6 BF  
49 3E F8 9D 18 DA FD

KEK=BA 3A 06 B2 C0 03 97 B1 02 DF 67 9E 5B 0C 9E 97 57 AB 34 8C 38  
50 82 E8 7D EF 86 21 A5 21 93 E3

#### 5.4.2. KDF-функція в групі точок еліптичної кривої (ECDH)

Функція формування ключа (KDF-функція) у циклічній групі простого поля (ECDHKDF) ґрунтується на міжнародних рекомендаціях RFC 3278, RFC 5008 та національному стандарті ДСТУ ISO/IEC 15946-3:2002 (Додаток А.3. Функція формування ключа ANSI X9.63).

5.4.2.1. Формування ключового матеріалу КМ виконується відповідно до алгоритму:

$$\text{КМ} = \text{H} (\text{ZZ} \parallel \text{counter} \parallel \text{SharedInfo} )$$

де

КМ – ключовий матеріал (рядок байтів), довжина якого залежить від алгоритму Н;

Н – геш-функція, визначається ідентифікатором алгоритму узгодження, що визначені у пункті 4.4.3.3, 4.4.3.4;

ZZ – спільний секрет, обчислений відповідно до пункту 5.3;

“counter” – це 32-х бітне число, яке представлено як бітовий вектор (чотири байти), записаного у зворотному порядку. Початковим значенням “counter” є 1 (одиниця) для будь-якого ZZ, а його бітовий вектор є “00 00 00 01” (hex); значення counter збільшується на одиницю (incremented) з кожним циклом виконання функції формування ключового матеріалу КМ (пункт 5.4.2.3) під час генерації ключа КШК;

SharedInfo – DER-кодована ASN.1 структура;

|| – означає операцію конкатенації.

#### 5.4.2.2. Структура “SharedInfo”

“SharedInfo” має таку ASN.1 структуру:

```
SharedInfo ::= SEQUENCE {  
    keyInfo AlgorithmIdentifier,  
    entityUInfo [0] EXPLICIT OCTET STRING OPTIONAL,  
    suppPubInfo [2] EXPLICIT OCTET STRING }
```

#### 5.4.2.3. Поля структури “SharedInfo”

“algorithm” – є ідентифікатор алгоритму ключа шифрування ключа (KeyWrapAlgorithm), на якому повинен бути зашифрований ключ шифрування повідомлення (даних). Параметри алгоритму повинні бути NULL (ASN.1 NULL).

“entityUInfo” – це випадковий рядок (аналогічний полю “partyAInfo” структури “OtherInfo”, наведеної у пункті 5.4.1.3), який генерує відправник. В CMS це значення розміщується в полі “ukm” (“UserKeyingMaterial”) (закодоване як OCTET STRING) структури “KeyAgreeRecipientInfo”. Довжина “partyAInfo” повинна бути 512 біт (64 байти);

“suppPubInfo” - це довжина сформованого ключа КШК в бітах (аналогічно полю “suppPubInfo” структури “OtherInfo”, наведеної у пункті 5.4.1.3), представлена як бітовий вектор (чотири байти) 32-бітного числа. Наприклад, ключ 192 біт повинен бути представлений як бітовий вектор “00 00 00 C0” (hex).

#### 5.4.1.4. Формування ключа КШК

Так як довжина ключового матеріалу (рядок байтів) КМ, сформованого відповідно до пункту 5.4.1.1, залежить від алгоритму Н, та може не бути рівною довжині ключа КШК (як рядка байтів), то використовується такий алгоритм:

А) Якщо довжина КМ більше, ніж довжина КШК, то за КШК приймають перші N байтів КМ, де N – довжина КШК.

Б) Якщо довжина КМ дорівнює довжині КШК, то за КШК приймають КМ.

В) Якщо довжина КМ менше, ніж довжина КШК, то:

Встановлюється значення counter = 1, та формується КМ1.

Встановлюється значення counter = 2, та виконується КМ2.

.....

(кількість кроків формування КМі визначається необхідною довжиною ключа КШК).

Отримані блоки ключового матеріалу об'єднуються (concatenation) для отримання необхідної довжини (останні байти останнього блоку КМі відкидаються):

$КМ = КМ1 \parallel КМ2 \parallel \dots$

Таким чином, що довжина КМ повинна бути рівною довжині ключа КШК.

#### 5.4.1.5. Щодо опціональності параметру “entityUInfo”

Якщо параметр “entityUInfo”, як не обов'язковий, не буде використовуватися, то у випадках А та Б для різних повідомлень буде формуватися один і той же ключ шифрування КШК. Для того щоб уникнути цього, у разі статичного механізму вимагається (а у разі динамічного – рекомендується) генерувати випадкове значення partyAInfo для кожного повідомлення, та використовувати під час формування КШК.

#### 5.4.2.5. Приклади обчислення ключа КШК

Примітка. В наведених прикладах у якості KeyWrapAlgorithm (у прикладах “wrapAlgorithm”) використовується ГОСТ 28147-87 у режимі гамування (“id-gost28147-ofb”, розділ 7) чи гамування із зворотнім зв'язком (“id-gost28147-cfb”, розділ 7). Ці алгоритми не є Wrap-алгоритмами (які викладені у розділ 6 далі), і використовуються тут лише для цілей наведення прикладів.

У прикладах використовується алгоритм узгодження з кофакторним множенням (пункт 4.4.3.4.1.) “id-dhSinglePass-cofactorDH-gost34311kdf-scheme”. Сформований ключ КШК позначається у прикладах через КЕК, його довжина в бітах – через keyLen.

ДКЕ для геш-функції ГОСТ 34.311 позначено через sBox (SBOX-1 – це ДКЕ №1 із переліку рекомендованих Інструкцією №114).

Приклад 1. ДСТУ 4145 ПБ, m=163, “id-gost28147-ofb” KeyWrapAlgorithm

sBox=SBOX-1  
curveOID=1.2.804.2.1.1.1.3.1.1.2.0  
wrapAlgorithm=1.2.804.2.1.1.1.1.1.2  
keyLen=256  
entityUInfo=0123456789abcdeffedcba98765432010123456789abcdeffedcba98  
765432010123456789abcdeffedcba98765432010123456789abcdeffedcba9876543201  
dA=00 03 04 99 1F 9A C1 A8 09 4F 6F BF A0 09 25 0D 4A 80 99 32 0D 55  
QAx=00 01 C5 6A 32 99 13 07 62 6D 09 B5 FF 06 9C 6C B8 0B 79 4D 39 BD  
QAy=00 01 B9 85 8C 28 B9 BC DC A1 7B A3 5E 6D 5F 03 4B 23 AA 74 33

C7

dB=00 01 FC 86 17 11 60 74 A8 FF 81 B4 2F 85 CA 25 16 CF 11 CE 2E 13  
QBx=00 01 F3 36 B1 E8 02 4B E4 AB E9 2D 26 CA 7F 30 22 0E 16 50 BC 1A  
QBy=00 02 F8 75 7B 1E 7E 72 E3 E5 46 B2 60 41 77 46 3D 3F C3 BE 63 C9  
ZZ=07 F8 4A DB A6 24 57 C5 DA 5D 95 94 47 C4 F6 C1 86 4C 9B 28 8E  
KEK=9F 61 94 11 BB 8D 53 C6 9C A7 C0 03 69 10 70 EF 31 C7 B7 2B 63 68  
31 10 33 AB EF B8 A0 C1 2C 9D

Приклад 2. ДСТУ 4145 ПБ, m=163, AES256Wrap KeyWrapAlgorithm

sBox=SBOX-1  
curveOID=1.2.804.2.1.1.1.3.1.1.2.0  
wrapAlgorithm=2.16.840.1.101.3.4.1.45  
keyLen=256  
entityUInfo=0123456789abcdeffedcba98765432010123456789abcdeffedcba98  
765432010123456789abcdeffedcba98765432010123456789abcdeffedcba9876543201  
dA=00 03 04 99 1F 9A C1 A8 09 4F 6F BF A0 09 25 0D 4A 80 99 32 0D 55  
QAx=00 01 C5 6A 32 99 13 07 62 6D 09 B5 FF 06 9C 6C B8 0B 79 4D 39 BD  
QAy=00 01 B9 85 8C 28 B9 BC DC A1 7B A3 5E 6D 5F 03 4B 23 AA 74 33

C7

dB=00 01 FC 86 17 11 60 74 A8 FF 81 B4 2F 85 CA 25 16 CF 11 CE 2E 13  
QBx=00 01 F3 36 B1 E8 02 4B E4 AB E9 2D 26 CA 7F 30 22 0E 16 50 BC 1A  
QBy=00 02 F8 75 7B 1E 7E 72 E3 E5 46 B2 60 41 77 46 3D 3F C3 BE 63 C9  
ZZ=07 F8 4A DB A6 24 57 C5 DA 5D 95 94 47 C4 F6 C1 86 4C 9B 28 8E  
KEK=74 D4 DB 20 7E 5F 44 76 B3 82 F4 CA 17 C7 4B 5D 2B FC A2 82 AD  
CE 72 76 B4 97 5C FD F0 59 75 F6

Приклад 3. ДСТУ 4145 ПБ, m=163, “id-gost28147-cfb” KeyWrapAlgorithm

sBox=SBOX-1  
curveOID=1.2.804.2.1.1.1.3.1.1.2.0  
wrapAlgorithm=1.2.804.2.1.1.1.1.1.3  
keyLen=256  
partyAInfo=  
dA=01 E7 C0 CE 88 C8 D4 44 E8 BA 58 70 90 F2 72 6D B1 BB 27 FB 58

QA<sub>x</sub>=06 A7 1C 9B 54 16 05 BA AD 2A 32 5A 2C 63 E3 90 F6 52 A1 B9 B4  
QA<sub>y</sub>=07 6D B8 D2 1D 01 41 D0 A9 E5 E5 BA 0C 66 A4 4A 23 A5 81 17 DB  
dB=03 04 99 1F 9A C1 A8 09 4F 6F BF A0 09 25 0D 4A 80 99 32 0D 55  
QB<sub>x</sub>=01 C5 6A 32 99 13 07 62 6D 09 B5 FF 06 9C 6C B8 0B 79 4D 39 BD  
QB<sub>y</sub>=01 B9 85 8C 28 B9 BC DC A1 7B A3 5E 6D 5F 03 4B 23 AA 74 33 C7  
ZZ=05 34 19 06 74 D9 3B 3D 23 27 D6 B0 65 20 7F 9D E7 80 63 65 CC  
KEK=6B CC 82 B8 F7 A8 BC 9F C8 B9 BD 4C DE 18 FC FE 99 71 17 55 D9  
AC 05 42 2C 33 32 F3 E9 6D 49 B8

Приклад 4. ДСТУ 4145 ОНБ, m=173, “id-gost28147-cfb” KeyWrapAlgorithm

```
algorithm=DSTU4145
sBox=SBOX-1
curveOID=1.2.804.2.1.1.1.3.1.2.2.0
wrapAlgorithm=1.2.804.2.1.1.1.1.1.3
keyLen=256
entityUInfo=0123456789abcdeffedcba98765432010123456789abcdeffedcba98
765432010123456789abcdeffedcba98765432010123456789abcdeffedcba9876543201
dA=12 A2 1B B4 69 32 46 54 43 58 0E 76 BD 13 DC EF 00 BA 4F 98 6A B9
QAx=17 58 84 26 EB 79 F8 C6 FA 82 2D D3 0A 9E 21 A7 7B 7D AE 5E CD
8D
QAy=0A D3 C9 C4 83 74 9D 9B 52 F0 89 63 00 7F 21 31 A6 91 17 40 96 A6
dB=12 A2 1B B4 69 32 46 54 43 58 0E 76 BD 13 DC EF 00 BA 4F 98 6A B9
QBx=17 58 84 26 EB 79 F8 C6 FA 82 2D D3 0A 9E 21 A7 7B 7D AE 5E CD
8D
QBy=0A D3 C9 C4 83 74 9D 9B 52 F0 89 63 00 7F 21 31 A6 91 17 40 96 A6
ZZ=03 AC 78 AD A3 0F C2 CB C8 F8 4E 64 F4 09 0F 81 6A F6 B6 B0 68 F0
KEK=4D 0A EE 86 1B 17 C2 5E AD 82 AA 16 54 44 95 BC 79 DE C9 8C 3D
7B 85 BF 3C 3B DD 2F DC C9 B9 ED
```

## **VI. Алгоритми захисту ключа шифрування даних “KeyWrapAlgorithm”**

У цій Технічній специфікації визначається алгоритм захисту ключа шифрування даних “KeyWrapAlgorithm”, що ґрунтується на міждержавному стандарті ГОСТ 28147-89 і міжнародних рекомендаціях RFC 2630, RFC 3394, та позначається як “GOST28147Wrap”.

### **6.1. Ідентифікатор алгоритму захисту “GOST28147Wrap”**

6.1.1. Ідентифікатор алгоритму захисту ключа шифрування даних “KeyWrapAlgorithm” вказуються в полі “EnvelopedData RecipientInfos KeyAgreeRecipientInfo keyEncryptionAlgorithm”. Ключ узгодження КШК формується за механізмами (протоколами) узгодження ключа ДН або ЕCDН:

KeyWrapAlgorithm ::= AlgorithmIdentifier

Обов'язковим для використання є алгоритм захисту "GOST28147Wrap".

### 6.1.3. Синтаксис "KeyWrapAlgorithm" алгоритму GOST28147Wrap

```
GOST28147Wrap ::= SEQUENCE {  
    algorithm      OBJECT IDENTIFIER,  
    parameters     GOST28147WrapParameters OPTIONAL }
```

6.1.2. Для алгоритму GOST28147Wrap поле "algorithm" ідентифікатора алгоритму "AlgorithmIdentifier" повинно містити об'єктний ідентифікатор "id-gost28147-wrap":

```
id-gost28147-wrap OBJECT IDENTIFIER ::= { iso(1) member-body(2)  
ukraine(804) root(2) security(1) cryptography(1) pki(1) pki-alg(1) pki-alg-sym(1)  
gost24147(1) wrap(5) }
```

6.1.3. Синтаксис поля "parameters" алгоритму алгоритму GOST28147Wrap такий:

```
GOST28147WrapParameters ::= CHOICE {  
    NULL,  
    parameters GOST28147Parameters  
},
```

```
GOST28147Parameters ::= SEQUENCE {  
    iv    OCTET STRING (SIZE (8)),  
    dke   OCTET STRING (SIZE (64))  
}
```

де

NULL - ASN.1 NULL;

"iv" – вектор ініціалізації, що обирається випадково

"dke" – довготривалий ключовий елемент (ДКЕ), відповідно до міждержавного стандарту ГОСТ 28147-89.

При використанні "GOST28147Wrap", як алгоритму захисту ключа шифрування ключів КШК в структурі "захищені дані" ("EnvelopedData"), параметри алгоритму повинні бути NULL. При цьому значення ДКЕ для алгоритму повинно братися із відкритого ключа одержувача.

Використання "GOST28147Wrap" з параметрами алгоритму, що не є NULL, не є предметом цієї Технічної специфікації.



## 6.2. Алгоритм GOST28147Wrap

### 6.2.1. Призначення алгоритму

Алгоритм GOST28147Wrap призначений для шифрування ключа (ключових даних чи інших даних, що підлягають захисту), використовуючи стандарт ГОСТ 28147-89 в режимі CFB (гамування із зворотнім зв'язком, відповідно до розділу 4 ГОСТ 28147-89, надалі - GOST28147-CFB), як стандарт шифрування.

Алгоритм GOST28147Wrap призначений також для забезпечення цілісності зашифрованих ключових даних.

### 6.2.2. Процес зашифрування (Key Wrap)

Вхідні дані процесу зашифрування:

- “iv” – вектор ініціалізації,
- “dke” – довготривалий ключовий елемент (ДКЕ),
- “КЕК” – ключ шифрування ключа (КШК),
- “plaintext” – ключові дані для зашифрування (у операції формування “захищені дані” – це ключ шифрування даних КШД).

Вихідні дані:

- “result” – зашифровані ключові дані.

Процес зашифрування виконується у такі кроки:

- 1) Виконати ініціалізацію алгоритму (особливості ініціалізаційних даних “iv” та “dke” наведено у пункті 6.2.4):
  - а). Встановити дані ініціалізації “iv” та “dke” як параметри (“parameters”) алгоритму GOST28147-CFB.
  - б). Виконати ініціалізацію алгоритму GOST28147-CFB, використовуючи параметри встановлені параметри алгоритму та ключ шифрування “КЕК”.

2) Обчислити контрольну суму ключових даних

Контрольна сума ключових даних (“checksum”) призначена для контролю правильності розшифрування зашифрованих ключових даних, та обчислюється як геш-значення ГОСТ 34.311 від ключового матеріалу. Значення “dke” для ГОСТ 34.311 береться те ж, що і на кроці 1.

З отриманого геш-значення беруться перші 8 байт як контрольна сума.

$checksum = \{ GOST34311(plaintext, dke) \} [8 \text{ bytes}]$ .

- 3) Виконати конкатенацію ключових даних з отриманою контрольною сумою:

`input = plaintext || checksum.`

4) Виконати зашифрування “input” ініціалізованим на кроці 1 алгоритмом:

`result = GOST28147-CFB_encrypt(input);`

Вихідними даними процесу зашифрування є “result”. Довжина вихідних даних “result” дорівнює довжині “plaintext”, збільшена на 8 байт контрольної суми.

У разі використання ГОСТ 28147-89 у якості алгоритму шифрування даних довжина “plaintext” становить 32 байти, а отже довжина “result” становить 40 байт.

### 6.2.3 Процес розшифрування (Key Unwrap)

Вхідні дані процесу розшифрування:

“result” – зашифровані ключові дані,

“iv” – вектор ініціалізації,

“dke” – довготривалий ключовий елемент (ДКЕ),

“КЕК” – ключ шифрування ключа (КШК).

Вихідні дані:

“plaintext” – ключові дані для зашифрування (у операції формування “захищені дані” – це ключ шифрування даних КШД).

Процес розшифрування виконується у такі кроки:

1) Виконати ініціалізацію алгоритму (особливості ініціалізаційних даних “iv” та “dke” наведено у пункті 6.2.4):

а). Встановити ініціалізаційні дані “iv” та “dke” як параметри (“parameters”) алгоритму GOST28147-CFB.

б). Виконати ініціалізацію алгоритму GOST28147-CFB, використовуючи параметри встановлені параметри алгоритму та ключ шифрування “КЕК”.

2) Виконати розшифрування “result” ініціалізованим на кроці 1 алгоритмом:

`input = GOST28147-CFB_decrypt(result);`

Вихідними даними процесу розшифрування є “input”. Довжина вихідних даних “input” дорівнює довжині “plaintext”, збільшена на 8 байт контрольної суми.

У разі використання ГОСТ 28147-89 у якості алгоритму шифрування даних довжина “plaintext” становить 32 байти, а отже довжина “input” становить 40 байт.

3) Отримати із отриманих на кроці 2 вихідних даних “input” контрольну суму “checksum”, як останні 8 байт вихідних даних “input” та “plaintext” як ключові дані (як байти “input” за виключенням останніх 8 байт):

$input = plaintext \parallel checksum,$

4) Обчислити контрольну суму ключових даних

Обчислена контрольна сума ключових даних (“checksum\_1”) призначена для контролю правильності розшифрування зашифрованих ключових даних, та обчислюється як геш-значення ГОСТ 34.311 від ключового матеріалу “plaintext”.

Значення “dke” для ГОСТ 34.311 береться те ж, що і на кроці 1.

З отриманого геш-значення беруться перші 8 байт як контрольна сума.

$checksum\_1 = \{ GOST34311(plaintext, dke) \} [8 \text{ bytes}].$

5) Порівняти отриману на кроці 3 контрольну суму “checksum” з обчисленою на кроці 4 “checksum\_1”.

У разі нееквівалентності зазначених контрольних сум, припинити подальше оброблення як “помилка розшифрування ключа”.

У разі еквівалентності зазначених контрольних сум, повернути як результат оброблення отримане значення ключового матеріалу “plaintext”.

#### 6.2.4. Особливості ініціалізаційних даних для “EnvelopedData”

При використанні “GOST28147Wrap”, як алгоритму захисту ключа шифрування ключів КШК в структурі “захищені дані” (“EnvelopedData”), ініціалізаційні дані “iv” (вектор ініціалізації, що обирається випадково) та “dke” (довготривалий ключовий елемент) визначаються таким чином:

“iv” – це перші 8 байт значення поля “ukm” (що має довжину 64 байти) структури “KeyAgreeRecipientInfo”. Якщо поле “ukm”, як не обов’язкове, відсутнє, то повинно братися таке значення за умовчанням:

$byte \text{ iv}[8] = \{ 0x4a, 0xdd, 0xa2, 0x2c, 0x79, 0xe8, 0x21, 0x05 \};$

“dke” – це значення ДКЕ, отримане із параметрів алгоритму відкритого ключа одержувача. Якщо значення ДКЕ, як не обов’язкове поле, відсутнє в параметрах алгоритму відкритого ключа одержувача, то повинно братися за умовчанням значення ДКЕ №1 (SBOX-1), визначеного Інструкцією №114.

## **VII. Алгоритм захисту даних (повідомлення) “contentEncryptionAlgorithm”**

### 7.1. Об’єктні ідентифікатори алгоритмів ГОСТ 28147-89

У якості алгоритму захисту (шифрування) даних “contentEncryptionAlgorithm” структури “EncryptedContentInfo” можуть використовуватися алгоритми міждержавного стандарту ГОСТ 28147-89 у таких режимах:

“id-gost28147-ofb” (режим гамування, розділ 3 ГОСТ 28147-89), та  
“id-gost28147-cfb” (режим гамування із зворотнім зв’язком, розділ 4 ГОСТ 28147-89).

```
id-gost28147-ofb OBJECT IDENTIFIER ::= { iso(1) member-body(2)
ukraine(804) root(2) security(1) cryptography(1) pki(1) pki-alg(1) pki-alg-sym(1)
gost24147(1) ofb(2) }
```

```
id-gost28147-cfb OBJECT IDENTIFIER ::= { iso(1) member-body(2)
ukraine(804) root(2) security(1) cryptography(1) pki(1) pki-alg(1) pki-alg-sym(1)
gost24147(1) cfb(3) }
```

### 7.2. Параметри алгоритмів ГОСТ 28147-89

```
GOST28147Parameters ::= SEQUENCE {
    iv OCTET STRING (SIZE (8)),
    dke OCTET STRING (SIZE (64))
},
```

де

“iv” – вектор ініціалізації, що обирається випадково;

“dke” – довготривалий ключовий елемент (ДКЕ), відповідно до міждержавного стандарту ГОСТ 28147-89.

## **VIII. Сертифікат шифрування**

8.1. Сертифікат шифрування у цих Технічних специфікаціях призначаються для використання їх в алгоритмах узгодження ключа Діффі-Геллмана: DH – в циклічній групі простого поля та ECDH – в групі точок еліптичної кривої.

8.2. Сертифікат шифрування, призначений для алгоритму узгодження ключа Діффі-Геллмана в циклічній групі простого поля (DH), повинен бути сертифікатом відкритого ключа алгоритму ГОСТ 34.310-95.

8.3. Сертифікат шифрування, призначений для алгоритму узгодження ключа Діффі-Геллмана в групі точок еліптичної кривої (ECDH), повинен бути сертифікатом відкритого ключа алгоритму ДСТУ 4145-2002.

8.4. Формат сертифікату шифрування повинен відповідати формату сертифіката відкритого ключа, визначеному в розділі 1 Технічних специфікацій форматів представлення базових об'єктів національної системи електронного цифрового підпису, затверджених наказом Департаменту спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України та Державного департаменту з питань зв'язку та інформатизації Міністерства транспорту та зв'язку України від 11.09.2006 № 99/166.

#### 8.5. Розширення сертифікату шифрування “Використання ключа”

8.5.1. В розширенні “Використання ключа” (“KeyUsage”) повинно бути встановлено значення:

“узгодження ключа” (“keyAgreement”).

8.5.2. Якщо в розширенні “Використання ключа” (“KeyUsage”) встановлено значення “Узгодження ключа” (“keyAgreement”), то додатково може бути встановлено значення:

“тільки зашифрування” (“encipherOnly”)

“тільки розшифрування” (“decipherOnly”).

8.5.3. Якщо в розширенні “Використання ключа” (“KeyUsage”) встановлено значення “Узгодження ключа” (“keyAgreement”), то не повинно бути встановлено значення:

“не-зречення” (“nonRepudiation”);

“шифрування ключа” (“keyEncipherment”);

“підписування сертифікатів” (“keyCertSign”);

“підписування списків відкликаних сертифікатів” (“cRLSign”).

#### Додаток 1. Приклад ASN.1 структури “захищені дані”

```
SEQUENCE { -- contentInfo
  OBJECTIDENTIFIER = 1 2 840 113549 1 7 3 -- contentType = envelopedData
  CONTEXT_0 { -- content [0]
    SEQUENCE { -- envelopedData
      INTEGER = 2 -- version
      -- originatorInfo [0] absent

      SET { -- RecipientInfos ::= SET SIZE (1..MAX) OF RecipientInfo
        CONTEXT_1 { -- kari [1] KeyAgreeRecipientInfo
          INTEGER = 3 -- version
          CONTEXT_0 { -- originator [0] EXPLICIT OriginatorIdentifierOrKey
```

```

CONTEXT_1 { -- originatorKey [1] OriginatorPublicKey
  SEQUENCE { -- OriginatorPublicKey
    OBJECTIDENTIFIER = 1 2 840 10046 2 1 -- dhPublicNumber algorithm
    -- parameters absent
  }
  BIT_STRING = 44 B9 26 32 13 77 AD 88 CD F5 9F 4B 4D A9 6C FF
38 60 EB 84 AB 45 E6 A3 F4 E2 94 27 97 F0 8D 29
A5 EB 1F 21 91 68 58 39 C8 F2 49 D8 99 DB 48 A8
9E 47 A5 9E 06 BE B4 F4 A0 86 01 10 C4 50 FB B1
F5 31 88 12 7B 15 18 70 F8 72 08 65 4F 51 A7 A3
96 18 E8 79 B4 A6 6C F1 B7 7A 61 26 F6 AF 4D 34
42 22 DD 80 F3 C7 42 CE 6A 1C 8C A6 24 E9 54 6A
A0 67 B1 80 DE BB B0 C4 FE BC 45 4C D2 EC 35 74
    -- publicKey
  } -- end CONTEXT_1 -- originatorKey
} -- end originator [0] EXPLICIT OriginatorIdentifierOrKey

CONTEXT_1 { -- ukm
  OCTET_STRING =
A974C4E9AA79D3CE5C74A4EDA5DB65F5C037D681F10A935F24A1DB9796EE878B79DBE90
71123CE70248430720283D57D60D3D4F6A74D4CC2E089FACD5920A293
} -- end ukm

SEQUENCE { -- keyEncryptionAlgorithm
  OBJECTIDENTIFIER = 1 2 840 113549 1 9 16 3 5 -- id-alg-ESDH
  SEQUENCE { -- ESDH algorithm parameters
    OBJECTIDENTIFIER = 1 2 840 113549 1 9 16 3 6 -- id_alg_CMS3DESwrap
    NULL = -- CMS3DESwrap algorithm parameters
  }
} -- end keyEncryptionAlgorithm

SEQUENCE { -- recipientEncryptedKeys ::= SEQUENCE OF RecipientEncryptedKey
  SEQUENCE { -- RecipientEncryptedKey
    SEQUENCE { -- rid KeyAgreeRecipientIdentifier
      SEQUENCE { -- issuerAndSerialNumber
        SET { -- issuer Name
          SEQUENCE {
            OBJECTIDENTIFIER = 2 5 4 3
            PRINTABLESTRING = CarlDSS
          }
        } -- end issuer Name
      }
      INTEGER = 201 -- serialNumber
    } -- end rid KeyAgreeRecipientIdentifier

    OCTET_STRING =
97A21C9B1D72034CFA1FCEDAAE8549E10D3204978043CB00496036A7DD4B0EE5D6A87BB
A669497A7
    -- encryptedKey
  } -- end RecipientEncryptedKey

} -- end recipientEncryptedKeys
} -- end kari KeyAgreeRecipientInfo

```

```
} -- end RecipientInfos

SEQUENCE { -- encryptedContentInfo
  OBJECTIDENTIFIER = 1 2 840 113549 1 7 1 -- contentType = Data
  SEQUENCE { --contentEncryptionAlgorithm
    OBJECTIDENTIFIER = 1 2 840 113549 3 7 -- DES-EDE3-CBC
    OCTET_STRING = 37E77ED71617C8AC -- IV
  } -- end contentEncryptionAlgorithm
  CONTEXT_0 = -- encryptedContent
6AF2B89A5865B2ADF43AA031B2BDF7527AEB2BFB04770FE259C633BB05FD0CEA
  } -- end encryptedContentInfo
  -- absent: unprotectedAttrs
} -- end EnvelopedData
} -- end content [0]
} -- end contentInfo
```

Мартиненко Сергій Васильович,  
кандидат фізико-математичних наук,  
директор технічний, НВФ "БКП-консалтинг"  
м.Київ, Україна; тел.: (044) 244-06-39, 465-28-19  
[martyn@itsway.kiev.ua](mailto:martyn@itsway.kiev.ua)