НАЦІОНАЛЬНИЙ СТАНДАРТ УКРАЇНИ

Інформаційні технології

МЕТОДИ ЗАХИСТУ КЕРУВАННЯ КЛЮЧАМИ

Частина 3. Механізми із застосуванням асиметричних методів (ISO/IEC 11770-3:1999, IDT)

ДСТУ ISO/IEC 11770-3:2002

Видання офіційне

БЗ № 4–2002/201

Київ
ДЕРЖСПОЖИВСТАНДАРТ УКРАЇНИ
2007

ПЕРЕДМОВА

1 ВНЕСЕНО: Державний науково-дослідний інститут технологій товарно-грошового обігу, фінансових і фондових ринків «УКРЕЛЕКОН» та Технічний комітет «Банківські і фінансові системи та технології» (ТК 105)

ПЕРЕКЛАД І НАУКОВО-ТЕХНІЧНЕ РЕДАГУВАННЯ: **М. Карнаух; М. Коваленко,** канд. техн. наук; **А. Нікітін,** д-р техн. наук

2 НАДАНО ЧИННОСТІ: наказ Держспоживстандарту України від 12 липня 2002 р. № 422 з 2003–10–01 зі зміною дати чинності згідно з наказом № 273 від 27 вересня 2005 р.

3 Ця частина стандарту відповідає ISO/IEC 11770-3:1999 Information technology — Security techniques — Key management — Part 3: Mechanisms using asymmetric techniques (Інформаційні технології. Методи захисту. Керування ключами.Частина 3. Механізми із застосуванням асиметричних методів)

Ступінь відповідності — ідентичний (IDT)

Переклад з англійської (en)

4 УВЕДЕНО ВПЕРШЕ

ЗМІСТ

НАЦІОНАЛЬНИЙ ВСТУП

Цей стандарт є тотожний переклад ISO/IEC 11770-3:1999 Information technology — Security techniques — Key management — Part 3: Mechanisms using asymmetric techniques (Інформаційні технології. Методи захисту. Керування ключами. Частина 3. Механізми із застосуванням асиметричних методів).

Технічний комітет, відповідальний за цей стандарт в Україні, — ТК 105 «Банківські і фінансові системи та технології».

Увага! Існує ймовірність, що деякі елементи цієї частини ISO/IEC 11770 можуть бути об'єктами патентного права. ISO і IEC не несуть відповідальності за зазначення будь-якого або всіх таких патентних прав.

До стандарту внесено такі редакційні зміни:

— слова «цей міжнародний стандарт» замінено на «цей стандарт»;

— до розділу «Нормативні посилання» долучено «Національне пояснення», виділене рамкою;

— структурні елементи стандарту: «Обкладинку», «Передмову», «Національний вступ», «Терміни та визначення понять» та «Бібліографічні дані» — оформлено згідно з вимогами національної стандартизації України.

ISO/IEC 11770 складається з таких частин під загальною назвою «Information technology — Security techniques — Key management»:

— Part 1: Framework.

— Part 2: Mechanisms using symmetric techniques.

— Part 3: Mechanisms using asymmetric techniques.

Додатки від A до E даної частини ISO/IEC 11770 необов'язкові і надані лише для інформації.

Перелік міжнародних стандартів, посилання на які є в ISO/IEC 11770-3, наведено в додатку D.

В Україні замість ISO/IEC 9798-3:1998 Information technology — Security techniques — Entity authentication mechanisms — Part 3: Mechanisms using digital signature techniques чинний ДСТУ ISO 9798-3:2002 Інформаційні технології. Методи захисту. Автентифікація суб'єктів. Частина 3. Механізми з використанням методу цифрового підпису.

Копії стандартів, на які є посилання, можна замовити в Головному фонді нормативних документів.

НАЦІОНАЛЬНИЙ СТАНДАРТ УКРАЇНИ

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

МЕТОДИ ЗАХИСТУ КЕРУВАННЯ КЛЮЧАМИ

Частина 3. Механізми із застосуванням асиметричних методів

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

МЕТОДЫ ЗАЩИТЫ УПРАВЛЕНИЕ КЛЮЧАМИ

Часть 3. Механизмы, использующие асимметричные методы

INFORMATION TECHNOLOGY

SECURITY TECHNIQUES KEY MANAGEMENT

Part 3. Mechanisms using asymmetric techniques

Чинний від 2006–10–01

1 СФЕРА ЗАСТОСУВАННЯ

Ця частина стандарту визначає механізми керування ключами, базовані на асиметричних криптографічних методах. А саме, вона стосується використання асиметричних методів для досягнення таких цілей:

1. Встановити розподілений таємний ключ для симетричних методів узгодженням ключа між двома суб'єктами *A* і *B*. Таємний ключ у механізмі узгодження таємного ключа є результатом обмінів даними між двома суб'єктами *A* і *B*. Жоден із них не може попередньо визначити значення розподіленого таємного ключа.

2. Встановити розподілений таємний ключ для симетричних методів передаванням ключа. Таємний ключ, у механізмі пересилання таємного ключа, обирає один суб'єкт *A* і пересилає іншому суб'єкту *B*, відповідним чином захищеним асиметричними методами.

3. Зробити1 відкритий ключ суб'єкта доступним для іншого суб'єкта передаванням ключа. У механізмі передавання відкритого ключа, відкритий ключ суб'єкта *A* треба передавати іншим суб'єктам із забезпеченням автентифікації, але нетаємно.

Деякі з механізмів цієї частини стандарту базовані на відповідних механізмах автентифікації ISO/IEC 9798-3.

Ця частина стандарту не стосується таких аспектів керування ключами, як:

— керування життєвим циклом ключа;

— механізмів генерації або верифікації пар асиметричних ключів;

— механізмів збереження, архівування, вилучення, знищення ключів тощо.

Незважаючи на те, що ця частина стандарту не стосується розподілення особистих ключів суб'єктів (із пар асиметричних ключів) від третьої довірчої сторони до суб'єкта-замовника, описані механізми передавання ключа можуть бути використані і для цього стандарту.

Ця частина стандарту не стосується впровадження перетворень, які використовують у керуванні ключами.

Примітка. Для досягнення автентичності повідомлень керування ключами можна забезпечити автентичність у протоколі встановлення ключа або використати систему підпису відкритим ключем для підписування повідомлень обміну ключами.

2 НОРМАТИВНІ ПОСИЛАННЯ

Наведені нижче стандарти містять положення, які через посилання в цьому тексті становлять положення цієї частини стандарту. На час опублікування цієї частини стандарту зазначені стандарти були чинними.

Усі стандарти підлягають перегляду, і учасників угод, базованих на цій частині стандарту, запрошують визначити можливість застосування найновіших видань стандартів, наведених нижче. Члени ISO та IEC впорядковують каталоги чинних міжнародних стандартів.

ISO 7498-2:1989 Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture

ISO/IEC 9594-8:1995 Information technology — Open Systems Interconnection — The Directory: Authentication framework

ISO/IEC 9798-3:1998 Information technology — Security techniques — Entity authentication — Part 3: Mechanisms using digital signature techniques

ISO/IEC 10118-1:1994 Information technology — Security techniques — Hash-functions — Part 1: General

ISO/IEC 10181-1:1996 Information technology — Open Systems Interconnection — Security frameworks for open systems: Overview

ISO/IEC 11770-1:1996 Information technology — Security techniques — Key management — Part 1: Framework.

НАЦІОНАЛЬНЕ ПОЯСНЕННЯ

ISO 7498-2: 1989 Системи оброблення інформації. Взаємозв'язок відкритих систем. Базова еталонна модель. Частина 2. Архітектура безпеки

ISO/IEC 9594-8:1995 Інформаційні технології. Взаємозв'язок відкритих систем. Довідник. Схема автентифікації

ISO/IEC 9798-3:1998 Інформаційні технології. Методи захисту. Автентифікація суб'єктів. Частина 3. Механізми з використанням методу цифрового підпису

ISO/IEC 10118-1:1994 Інформаційні технології. Методи захисту. Геш-функції. Частина 1. Загальні положення

ISO/IEC 10181-1:1996 Інформаційні технології. Взаємозв'язок відкритих систем. Схеми безпеки для відкритих систем. Частина 1. Схеми

ISO/IEC 11770-1: 1996 Інформаційні технології. Методи захисту. Керування ключами. Частина 1. Загальні положення.

3 ТЕРМІНИ ТА ВИЗНАЧЕННЯ ПОНЯТЬ І ПОЗНАКИ

У цій частині стандарту застосовують такі терміни:

3.1 асиметричний криптографічний метод (*asymmetric cryptographic technique*)

Криптографічний метод, який використовує два пов'язані між собою перетворення — відкрите перетворення (визначене відкритим ключем) і приватне перетворення (визначене особистим ключем). Ці два перетворення мають таку властивість, що за наданим відкритим перетворенням неможливо в результаті обчислень вивести приватне перетворення.

Примітка. Система, заснована на асиметричному криптографічному методі, може бути або шифрувальною системою, системою цифрового підпису, комбінованою системою — шифрувальною і цифрового підпису, або системою узгодження ключів. В асиметричному криптографічному методі є чотири елементарних перетворення: підписування і верифікація для систем цифрового підпису, зашифровування і розшифровування для шифрувальних систем. Перетворення підписування і розшифровування зберігає особисто суб'єкт-власник, тоді як відповідне перетворення верифікації і розшифровування оголошують. Існують асиметричні криптосистеми (наприклад RSA), в яких означені чотири елементарні функції можна виконати лише двома перетвореннями: одне приватне перетворення достатнє як для підписування, так і для розшифровування повідомлень, і одне відкрите перетворення достатнє як для верифікації, так і для зашифровування повідомлень. Однак, оскільки це не задовольняє принципу розмежування ключів, в цій частині стандарту чотири елементарні перетворення і відповідні ключі розглядають окремо

3.2 система асиметричного шифрування (*asymmetric encipherment system*)

Система, базована на асиметричних криптографічних методах, відкрите перетворення якої використовують для зашифровування, а приватне перетворення — для розшифровування

3.3 асиметрична пара ключів (*asymmetric key pair*)

Пара взаємопов'язаних ключів, де особистий ключ визначає приватне перетворення, а відкритий ключ — відкрите перетворення

3.4 повноважний сертифікатор (*certification authority;* CA)

Центр, якому довірено створювати і призначати сертифікати відкритих ключів. Необов'язково, повноважний сертифікатор може створювати і призначати ключі суб'єктам

3.5 криптографічна контролювальна функція (*cryptographic check function*)

Криптографічне перетворення, яке одержує на вході таємний ключ і довільний рядок і видає на виході криптографічне контролювальне значення. Обчислювання правильного контролювального значення без знання таємного ключа повинне бути неможливим (ISO/IEC 9798-1)

3.6 криптографічне контролювальне значення (*cryptographic check value*)

Інформація, виведена виконуванням криптографічного перетворювання одиниці даних (ISO/IEC 9798-4)

3.7 розшифровування (*decipherment*)

Перетворювання, обернене до відповідного зашифровування (ISO/IEC 11770-1)

3.8 цифровий підпис (*digital signature*)

Дані, додавані до одиниці даних або криптографічне перетворювання одиниці даних, яке дає змогу одержувачу одиниці даних довести походження і цілісність одиниці даних і захистити посилача і одержувача одиниці даних від їх підроблення третьою стороною, а також посилача від підроблення одержувачем

3.9 розрізнювальний ідентифікатор (*distinguishing identifier*)

Інформація, яка однозначно розрізняє суб'єктів (ISO/IEC 11770-1)

3.10 зашифровування (*encipherment*)

(оборотне) перетворювання даних за криптографічним алгоритмом для вироблення шифротексту, тобто для приховання інформаційного змісту даних (ISO/IEC 11770-1)

3.11 автентифікація суб'єкта (*entity authentication*)

Підтвердження тотожності заявленого суб'єкта (ISO/IEC 9798-1)

3.12 автентифікація суб'єкта *A* для *B* (*entity authentication of A to B*)

Засвідчення тотожності суб'єкта *A* для суб'єкта *B*

3.13 явна автентифікація ключа від *A* для *B* (*explicit key authentication from A to B*)

Засвідчення суб'єкту *B*, що *A* є єдиним іншим суб'єктом, який володіє правильним ключем.

Примітка. Неявна автентифікація ключа від А до В і підтвердження ключа від А до В разом запроваджують явну автентифікацію ключа від А до В

3.14 неявна автентифікація ключа від *A* для *B* (*implicit key authentication from A to B*)

Засвідчення суб'єкту *B*, що *A* є єдиним іншим суб'єктом, який імовірно може володіти правильним ключем

3.15 ключ (*key*)

Послідовність символів, яка контролює виконання криптографічного перетворювання (наприклад, зашифровування, розшифровування, обчислювання криптографічної контролювальної функції, обчислювання підпису або верифікацію цього підпису) (ISO/IEC 11770-1)

3.16 узгоджування ключів (*key agreement*)

Процес встановлювання розподіленого таємного ключа між суб'єктами таким чином, що жоден із них не може заздалегідь визначити значення ключа

3.17 підтвердження ключа від *A* для *B* (*key confirmation from A to B*)

Засвідчення суб'єкту *B*, що суб'єкт *A* володіє правильним ключем

3.18 контролювання ключа (*key control*)

Спроможність вибрати ключ або параметри, використовувані для його обчислювання

3.19 встановлювання ключа (*key establishment*)

Процес надавання доступу одному або кільком суб'єктам до розподіленого таємного ключа. Встановлювання ключа охоплює його погодження і передавання

3.20 маркер ключа (*key token*)
Повідомлення керування ключами, яке надсилають від одного суб'єкта до іншого під час виконання механізму керування ключами

3.21 передавання ключа (*key transport*)
Відповідним чином захищений процес передавання ключа від одного суб'єкта до іншого

3.22 взаємна автентифікація суб'єктів (*mutual entity authentication*)
Автентифікація суб'єкта, яка забезпечує обом суб'єктам засвідчення тотожності іншого

3.23 односпрямована функція (*one-way function*)
Функція з такою властивістю, що легко обчислити вихідні дані для наданих вхідних даних, але неможливо в результаті обчислень знайти для наданих вихідних даних вхідні, які відображаються в ці вихідні дані

3.24 особистий ключ (*private key*)
Ключ з асиметричної пари ключів суб'єкта, який може бути використаний лише цим суб'єктом
Примітка. У випадку асиметричної системи підпису особистий ключ визначає перетворення підпису. У випадку асиметричної системи шифрування особистий ключ визначає перетворення розшифровування

3.25 відкритий ключ (*public key*)
Ключ з асиметричної пари ключів суб'єкта, який може бути загальнодоступним.
Примітка. У випадку асиметричної системи підпису відкритий ключ визначає перетворення верифікації. У випадку асиметричної системи шифрування відкритий ключ визначає перетворення зашифровування. Ключ, який є «загальновідомим», необов'язково є доступним для всіх. Ключ може бути доступним лише усім членам заздалегідь визначеної групи

3.26 сертифікат відкритого ключа (*public key certificate*)
Інформація щодо відкритого ключа суб'єкта, підписана повноважним сертифікатором, внаслідок чого вона стає непідробною

3.27 інформація щодо відкритого ключа (*public key information*)
Інформація, яка містить, щонайменше, розрізнювальний ідентифікатор і відкритий ключ. Інформація щодо відкритого ключа обмежена датою стосовно одного суб'єкта і одним відкритим ключем цього суб'єкта. В інформації щодо відкритого ключа може розміщуватися інша постійна інформація стосовно повноважного сертифікатора, суб'єкта, відкритого ключа, обмежень на використання ключа, строку дійсності або задіяного алгоритму

3.28 таємний ключ (*secret key*)
Ключ, який використовує у симетричних криптографічних методах певна множина суб'єктів

3.29 порядковий номер (*sequence number*)
Змінюваний з часом параметр, значення якого обирають із означеної послідовності і яке не повторюється в межах певного строку (ISO/IEC 11770-1)

3.30 система підпису (*signature system*)
Система, базована на асиметричних криптографічних методах, приватне перетворення якої використовують для підписування, а відкрите перетворення використовують для верифікації

3.31 позначка часу (*time stamp*)
Одиниця даних, яка означує момент часу по відношенню до загального еталону часу

3.32 повноважний позначальник часу (*time stamping authority*)
Третя довірча сторона, якій довіряють надавати свідоцтво, яке містить час генерування надійної позначки часу (ISO/IEC 13888-1)

3.33 змінний з часом параметр (*time variant parameter*)
Одиниця даних, така як випадкове число, порядковий номер або позначка часу, яку використовують для верифікації того, що повідомлення неповторне

3.34 третя довірча сторона (*trusted third party*)
Повноважний з безпеки або його агент, якому інші суб'єкти довіряють стосовно діяльності, пов'язаної з безпекою (ISO/IEC 10181-1).

4 СИМВОЛИ ТА СКОРОЧЕННЯ

У цій частині стандарту використовують такі символи та скорочення:

A, B — розрізнювальні ідентифікатори суб'єктів;
BE — зашифрований блок даних;
BS — підписаний блок даних;
CA — повноважний сертифікатор;
$Cert_A$ — сертифікат відкритого ключа суб'єкта A;
D_A — приватне перетворення розшифровування суб'єкта A;
d_A — приватний ключ розшифровування суб'єкта A;
E_A — відкрите перетворення зашифровування суб'єкта A;
e_A — відкритий ключ зашифровування суб'єкта A;
$F(h, g)$ — функція узгодження ключа;
f — криптографічна контролювальна функція;
$f_K(Z)$ — криптографічне контролювальне значення, яке є результатом застосування криптографічної контролювальної функції f із використанням на вході таємного ключа K та довільного рядка даних Z;
g — спільний елемент, відкрито розділюваний усіма суб'єктами, які використовують функцію узгодження ключа F;
h_A — особистий ключ узгодження ключа суб'єкта A;
$hash$ — геш-функція;
H — множина елементів;
G — множина елементів;
K — таємний ключ для симетричної криптографічної системи;
K_{AB} — таємний ключ, розділюваний суб'єктами A і B;

Примітка. У запровадженнях на практиці таємний розподілений ключ має підлягати додатковому обробленню перш, ніж зможе бути використаним для симетричної криптографічної системи.

KT — маркер ключа;
KT_{Ai} — маркер ключа, надісланий суб'єктом A після виконання стадії i;
p_A — відкритий ключ узгодження ключа суб'єкта A;
PKI_A — інформація щодо відкритого ключа суб'єкта A;
r — випадкове число, згенероване під час роботи механізму;
r_A — випадкове число, видане суб'єктом A в механізмі узгодження ключа;
S_A — приватне перетворення підписування суб'єкта A;
s_A — особистий ключ підпису суб'єкта A;
$Text\ i$ — необов'язкове поле даних, яке використовують відповідно до сфери застосування цієї частини стандарту;
TVP — змінюваний з часом параметр такий, як випадкове число, позначка часу або порядковий номер;
V_A — відкрите перетворення верифікації суб'єкта A;
v_A — відкритий ключ верифікації суб'єкта A;
ω — односпрямована функція;
Σ — цифровий підпис;
$\|$ — конкатенація двох елементів даних.

Примітка 1. Щодо природи перетворення підписування не робиться жодних припущень. У випадку системи підписування з відновленням повідомлення, $S_A(m)$ позначає безпосередньо підпис Σ. У випадку системи підписування з додатком, $S_A(m)$ позначає повідомлення m разом із підписом Σ.

Примітка 2. Ключі асиметричної криптографічної системи позначають малою літерою (яка зазначає функцію того ключа) з індексом, який є ідентифікатором його власника, наприклад, відкритий ключ верифікації, що належить суб'єкту A, позначають v_A. Відповідні перетворення позначають великими літерами з індексом, який є ідентифікатором їх власника, наприклад, відкрите перетворення верифікації суб'єкта A позначають V_A.

5 ВИМОГИ

Припускаємо, що суб'єкти обізнані у заявлених тотожностях одне одного. Це може бути досягнуто розміщенням ідентифікаторів в інформації, якою обмінюються два суб'єкти, або це може бути очевидним із контексту використовування механізму. Верифікація ідентичності означає кон-

троль відповідності одержаного поля ідентифікатора деякому відомому (довірчому) значенню або попереднім очікуванням.

Якщо відкритий ключ зареєстровано за деяким суб'єктом, цей суб'єкт повинен впевнитися, що суб'єкт, який реєструє ключ, володіє відповідним особистим ключем (див. частину 1 цього стандарту стосовно реєстрації ключа).

6 УЗГОДЖЕННЯ ТАЄМНОГО КЛЮЧА

Узгодження ключа — це процес встановлювання розподіленого таємного ключа між двома суб'єктами в такий спосіб, що жоден із них не може попередньо визначити значення розподіленого таємного ключа. Механізм узгодження ключа може забезпечити неявну автентифікацію ключа; в контексті встановлення ключа неявна автентифікація ключа означає, що після виконання механізму лише ідентифікований суб'єкт може володіти правильним розподіленим таємним ключем.

Узгодження ключа між двома суб'єктами A і B має місце в контексті, який розподіляють два суб'єкти. Контекст складається з об'єктів: множина G, множина H і функція F. Функція F повинна задовольняти такі вимоги:

— F виконується над двома вхідними даними: один елемент h з H і другий елемент g з G, і видає результат y в G, $y = F(h, g)$.

— F задовольняє умови комутативності $F(h_A, F(h_B, g)) = F(h_B, F(h_A, g))$.

— Методом обчислювань неможливо віднайти $F(h_1, F(h_2, g))$ з $F(h_1, g)$, $F(h_2, g)$ і g. Це означає, що $F(*, g)$ є односпрямованою функцією.

— Суб'єкти A і B розподіляють спільний елемент g з G, який може бути загальновідомим.

— Суб'єкти, які діють за цих умов можуть ефективно обчислювати значення функції $F(h,g)$ і ефективно генерувати випадкові елементи з H.

Залежно від конкретного механізму узгодження ключа, можуть бути накладені додаткові умови.

Примітка 1. Приклад можливої функції F наведено в додатку В.

Примітка 2. У запровадженнях на практиці, таємний розподілений ключ може підлягати додатковому оброблянню. Виведений розподілений таємний ключ може бути обчислений 1) безпосереднім виділенням бітів із розподіленого таємного ключа K_{AB} або 2) перетворенням розподіленого таємного ключа K_{AB} і, необов'язково, інших нетаємних даних односпрямованою функцією і виділення бітів на її вихідних даних.

Примітка 3. У загальному випадку, необхідно контролювати отримані значення функції $F(h,g)$ на слабкі значення. Якщо такі значення виявлені, робота протоколу повинна бути припинена. Приклад, відомий як узгодження ключа за Діффі-Гелманом, подано в В.5.

6.1 Узгодження ключа, механізм 1

Цей механізм узгодження ключа не інтерактивно, з взаємною неявною автентифікацією ключа встановлює розподілений таємний ключ між суб'єктами A і B. Повинні задовольнятися такі вимоги:

1. Кожен суб'єкт X має особистий ключ узгодження ключа h_X в H і відкритий ключ узгодження ключа $p_X = F(h_x, g)$.

2. Кожен суб'єкт має доступ до автентифікованої копії відкритого ключа узгодження ключа іншого суб'єкта. Цього можна досягти з використанням механізму розділу 8.

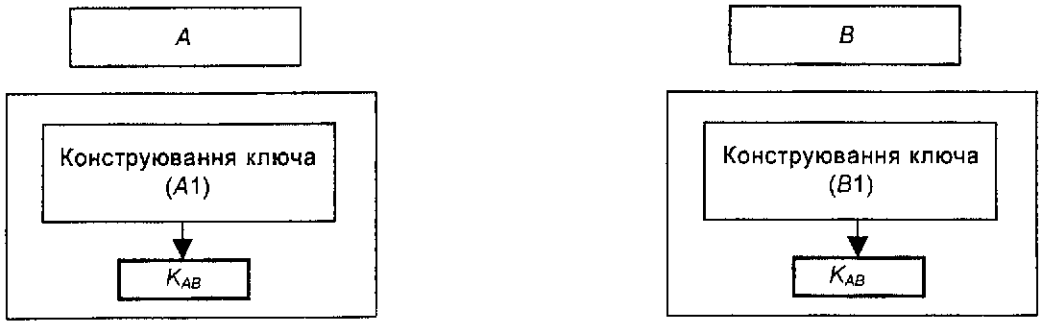

Рисунок 1 — Узгодження ключа, механізм 1

Конструювання ключа (A1)

З використанням свого власного особистого ключа узгодження ключа h_A і відкритого ключа узгодження ключа p_B, який належить B, A обчислює розподілений таємний ключ так:

$$K_{AB} = F(h_A, p_B).$$

Конструювання ключа (*B*1)

З використанням свого власного особистого ключа узгодження ключа h_B і відкритого ключа узгодження ключа p_A, який належить *A*, *B* обчислює розподілений таємний ключ так

$$K_{AB} = F(h_B, p_A).$$

Як наслідок вимоги 2 до *F* обидва обчислені значення K_{AB} є тотожні.

Примітка. Цей механізм узгодження ключа має такі властивості:

1. Число проходів 0. Як наслідок, таємний розподілений ключ завжди має те саме значення (див. розділ 6, примітка 2).
2. Автентифікація ключа: механізм забезпечує взаємну неявну автентифікацію ключа.
3. Підтвердження ключа: механізм не забезпечує підтвердження ключа.
4. Це механізм узгодження ключа внаслідок того, що встановлений ключ є односпрямованою функцією особистого ключа узгодження ключа h_A і h_B для *A* і *B* відповідно. Однак, один суб'єкт може знати відкритий ключ іншого перед тим, як обрати їх особистий ключ. Такий суб'єкт може вибрати орієнтовно *s* бітів ключа, що встановлюється, за рахунок генерування 2^s пробних значень для їх особистого ключа узгодження ключа за час від визначення відкритого ключа іншого суб'єкта до вибору їх власного особистого ключа.
5. *Приклад*: приклад, відомий як узгодження ключа за Діффі-Гелманом, подано в В.5.

6.2 Узгодження ключа, механізм 2

Цей механізм узгодження ключа за один прохід установлює розподілений таємний ключ між суб'єктами *A* і *B* з неявною автентифікацією ключа від *B* до *A*, але без автентифікації від *A* до *B* (тобто *B* не знає з ким він встановив розподілений таємний ключ). Повинні задовольнятися такі вимоги:

1. Суб'єкт *B* має особистий ключ узгодження ключа h_B в *H* і відкритий ключ узгодження ключа $p_B = F(h_B, g)$.

2. Суб'єкт *A* має доступ до автентифікованої копії відкритого ключа узгодження ключа *B* — p_B. Цього можна досягти з використанням механізму розділу 8.

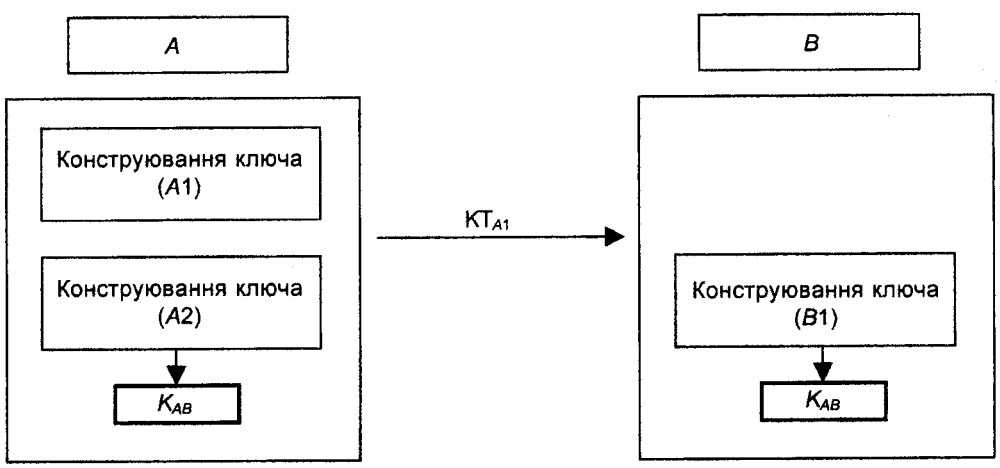

Рисунок 2 — Узгодження ключа, механізм 2

Конструювання маркера ключа (*A*1)

A випадково і таємно генерує *r* в *H*, обчислює *F(r, g)* і надсилає до *B* маркер ключа

$$KT_{A1} = F(r, g) \,||\, Text.$$

Конструювання ключа (*A*2)

Додатково, *A* обчислює ключ, як

$$K_{AB} = F(r, p_B).$$

Конструювання ключа (*B*1)

B виділяє *F(r, g)* з одержаного маркера KT_{A1} і обчислює розподілений таємний ключ

$$K_{AB} = F(h_B, F(r, g)).$$

Відповідно до вимоги 1 щодо *F* обидва обчислені значення K_{AB} є тотожні.

Примітка. Цей механізм узгодження ключа має такі властивості:

1. Число проходів: 1.

2. Автентифікація ключа: механізм забезпечує неявну автентифікацію ключа від *B* до *A* (*B* є єдиним суб'єктом, крім *A*, який може обчислити розподілений таємний ключ).

3. Підтвердження ключа: механізм не реалізує підтвердження ключа.

4. Це механізм узгодження ключа внаслідок того, що встановлений ключ є односпрямованою функцією випадкового числа *r*, постаченого особистими ключами узгодження ключа суб'єктів *A* і *B*. Однак, внаслідок того, що суб'єкт *A* може знати відкритий ключ суб'єкта *B* перед тим, як обрати значення *r*, *A* може вибрати орієнтовно *s* бітів ключа, що встановлюється, за рахунок генерування 2^s пробних значень для *r* за час від визначання відкритого ключа *B* до надсилання KT_{A1}.

5. *Приклад*: приклад цього механізму узгодження ключа (відомий, як узгодження ключа за ЕльГамаль) наведено в В.3.

6. Використання ключа: внаслідок того, що *B* одержує ключ K_{AB} від неавтентифікованого суб'єкта *A*, безпечне використання K_{AB} на стороні *B* обмежене функціями, які не вимагають довіри до автентичності *A*, такими як розшифровування і генерування кодів автентифікації повідомлень.

6.3 Узгодження ключа, механізм 3

Цей механізм узгодження ключа за один прохід установлює розподілений таємний ключ між суб'єктами *A* і *B* із взаємною неявною автентифікацією ключа і автентифікацією суб'єкта *A* для *B*. Повинні задовольнятися такі вимоги:

1. Суб'єкт *A* має асиметричну систему підпису (S_A, V_A).

2. Суб'єкт *B* має доступ до автентифікованої копії відкритого перетворення верифікації V_A. Це може бути досягнуто використанням механізмів розділу 8.

3. Суб'єкт *B* має систему узгодження ключа з ключами (h_A, p_B).

4. Суб'єкт *A* має доступ до автентифікованої копії відкритого ключа узгодження ключів p_B суб'єкта *B*. Це може бути досягнуто використанням механізмів розділу 8.

5. *TVP*: *TVP* повинен бути або позначкою часу, або порядковим номером. Якщо використовують позначку часу, необхідні надійні і синхронізовані годинники, якщо використовують порядковий номер, необхідні можливість ведення та верифікації двосторонніх лічильників.

6. Суб'єкти *A* і *B* повинні узгодити криптографічну контролювальну функцію *f* (аналогічну стандартизованій в ISO/IEC 9797) і спосіб включення K_{AB} як ключа до цієї контролювальної функції.

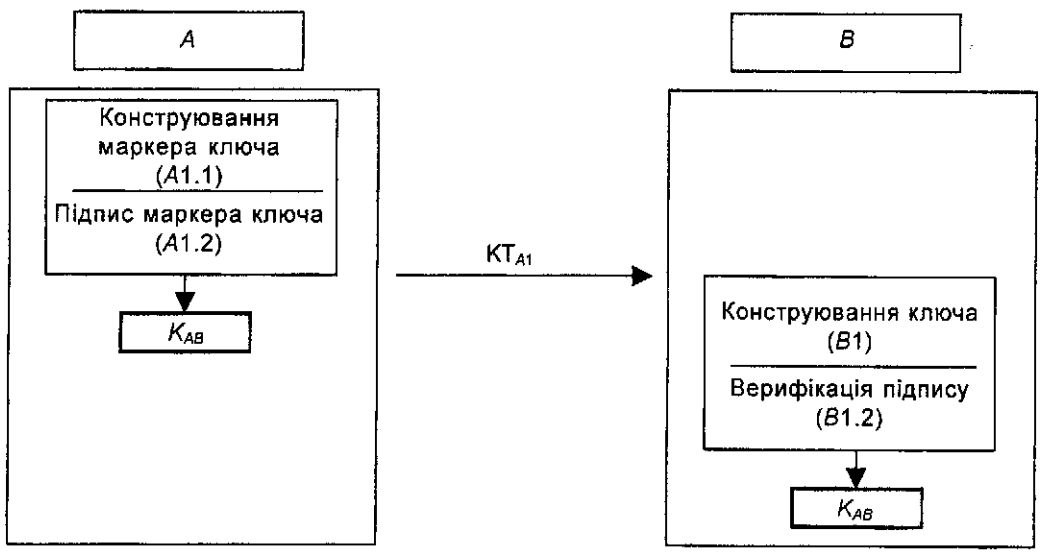

Рисунок 3 — Узгодження ключа, механізм 3

Конструювання ключа (*A*1.1)

A випадково і таємно генерує *r* і обчислює *F*(*r*, *g*). *A* обчислює розподілений таємний ключ як

$$K_{AB} = F(r, p_B).$$

Використовуючи розподілений таємний ключ K_{AB}, *A* обчислює криптографічне контролювальне значення об'єднанням розрізнювального ідентифікатора посилача *A* і порядкового номера або позначки часу *TVP*.

Конструювання маркера ключа (*A*1.2)

A підписує криптографічне контролювальне значення з використанням свого приватного перетворення підписування S_A. Потім *A* формує маркер ключа, який складається з розрізнюваль-

ного ідентифікатора посилача *A*, вхідних даних для ключа *F(r, g)*, *TVP*, підписаного криптографічного контролювального значення і деяких необов'язкових даних

$$KT_{A1} = A||F(r, g)||TVP||S_A(f_{KAB}(A||TVP))||Text1.$$

Надсилає його до *B*.

Конструювання ключа (*B*1.1)

B виділяє *F(r, g)* з одержаного маркера ключа і обчислює розподілений таємний ключ із використанням свого особистого ключа узгодження ключа h_B,

$$K_{AB} = F(h_B, F(r, g)).$$

Використовуючи розподілений таємний ключ K_{AB}, *B* обчислює криптографічне контролювальне значення на розрізнювальному ідентифікаторі посилача *A* і *TVP*.

Верифікація підпису (*B*1.2)

B використовує відкрите перетворення верифікації посилача V_A для верифікації підпису *A*, а з тим цілісність і походження одержаного маркера KT_{A1}. Потім *B* підтверджує неповторюваність у часі маркера (оглядом *TVP*).

Примітка. Цей механізм узгодження ключа має такі властивості:

1. Число проходів: 1.
2. Автентифікація ключа: цей механізм забезпечує явну автентифікацію ключа від *A* до *B* і неявну автентифікацію ключа від *B* до *A*.
3. Підтвердження ключа: цей механізм забезпечує підтвердження ключа від *A* до *B*.
4. Це механізм узгодження ключа внаслідок того, що встановлений ключ є односпрямованою функцією випадкового числа *r*, постаченого особистими ключами узгодження ключа суб'єктів *A* і *B*. Однак, внаслідок того, що суб'єкт *A* може знати відкритий ключ суб'єкта *B* перед тим, як обрати значення *r*, *A* може вибрати орієнтовно *s* бітів ключа, що встановлюється, за рахунок генерування 2^s пробних значень для *r* за час від визначання відкритого ключа *B* до надсилання KT_{A1}.
5. *TVP*: забезпечує автентифікацію суб'єкта *A* для *B* і запобігає повторенню маркера ключа.
6. *Приклад*: приклад цього механізму узгодження ключа (відомий як узгодження ключа за Ніберг-Рюппелем) наведений в В.4.
7. Сертифікати відкритого ключа: якщо *Text1* використовують для пересилання сертифіката відкритого ключа *A*, то вимога 2 на початку цього розділу може бути послаблена до вимоги, щоб *B* володів автентифікованою копією відкритого ключа верифікації СА.

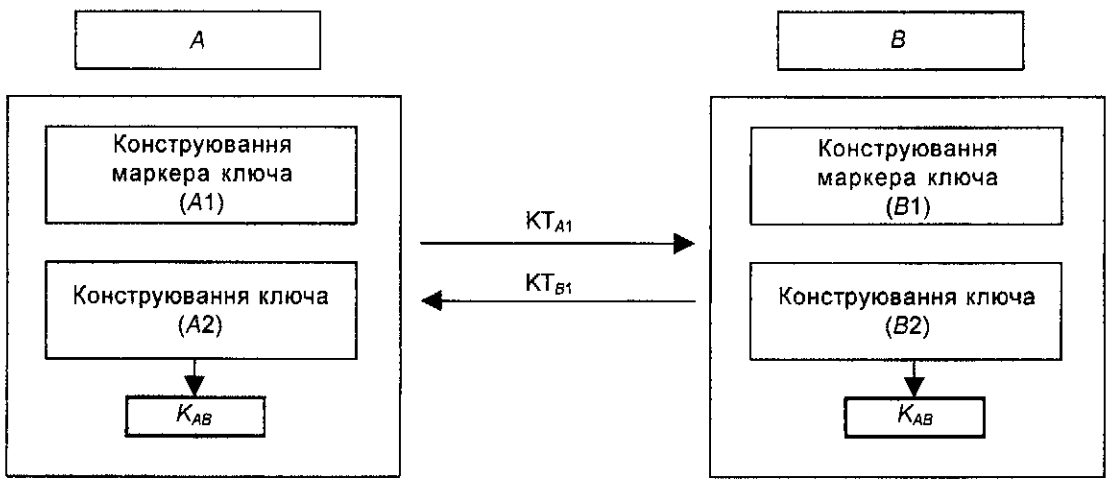

Рисунок 4 — Узгодження ключа, механізм 4

6.4 Узгодження ключа, механізм 4

Цей механізм узгодження ключа за два проходи встановлює розподілений таємний ключ між суб'єктами *A* і *B* зі спільним контролем ключа без попереднього обміну ключовою інформацією. Механізм не забезпечує ні автентифікації суб'єкта, ні автентифікації ключа.

Конструювання маркера ключа (*A*1)

A випадково і таємно генерує r_A в *H*, обчислює $F(r_A, g)$, конструює маркер ключа

$$KT_{A1} = F(r_A, g)||Text1$$

і надсилає його до *B*.

Конструювання маркера ключа (*B*1)

B випадково і таємно генерує r_B в *H*, обчислює $F(r_B, g)$, конструює маркер ключа

$$KT_{B1} = F(r_B, g) \,||\, Text2$$

і надсилає його до *A*.

Конструювання ключа (*A*2)

A виділяє $F(r_B, g)$ з одержаного маркера KT_{B1} і обчислює розподілений таємний ключ

$$KT_{AB} = F(r_A, F(r_B, g)).$$

Конструювання ключа (*B*2)

B виділяє $F(r_A, g)$ з одержаного маркера KT_{A1} і обчислює розподілений таємний ключ

$$KT_{AB} = F(r_B, F(r_A, g)).$$

Примітка. Цей механізм узгодження ключа має такі властивості:

1. Число проходів: 2.

2. Автентифікація ключа: цей механізм не забезпечує автентифікації ключа. Однак, цей механізм може бути корисним у середовищі, де автентичність маркера ключа верифікують із використанням інших засобів. Для прикладу, ґеш-кодом маркера ключа суб'єкти можуть обмінятися між собою з використанням другого каналу взаємодії. Дивись також механізм передавання відкритого ключа 2.

Підтвердження ключа: механізм не забезпечує підтвердження ключа.

3. Це механізм узгодження ключа внаслідок того, що встановлений ключ є односпрямованою функцією випадкових чисел r_A і r_B, постачених, відповідно, *A* і *B*. Однак, внаслідок того, що суб'єкт *B* може знати $F(r_A, g)$ перед тим, як обрати значення r_B, суб'єкт *B* може обрати орієнтовно *s* бітів ключа, що встановлюється, за рахунок генерації 2^s пробних значень r_B за час від одержання KT_{A1} до надсилання KT_{B1}.

4. *Приклад*: приклад цього механізму (відомий як узгодження ключа за Діффі-Гелманом) подано в В.5.

6.5 Узгодження ключа, механізм 5

Цей механізм узгодження ключа за два проходи встановлює розподілений таємний ключ між суб'єктами *A* і *B* із неявною взаємною автентифікацією ключа і спільним контролем ключа. Повинні задовольнятися такі вимоги:

1. Кожен суб'єкт *X* має особистий ключ узгодження ключа h_X в *H* і відкритий ключ узгодження ключа $p_X = F(h_X, g)$.

2. Кожен суб'єкт має доступ до автентифікованої копії відкритого ключа узгодження ключа іншого суб'єкта. Цього можна досягти використанням механізмів розділу 8.

3. Обидва суб'єкти повинні узгодити спільну односпрямовану функцію ω.

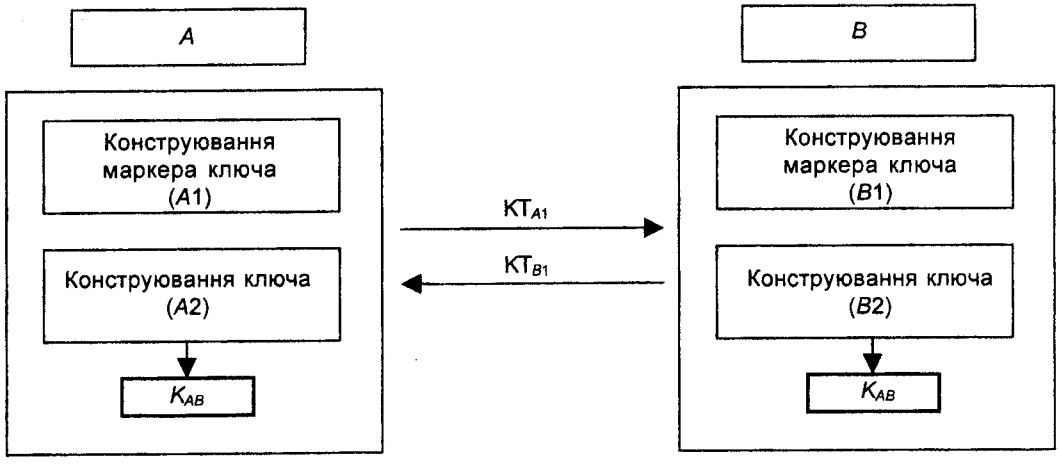

Рисунок 5 — Узгодження ключа, механізм 5

Конструювання маркера ключа (*A*1)

A випадково і таємно генерує r_A в *H*, обчислює $F(r_A, g)$ і надсилає до *B* маркер ключа

$$KT_{A1} = F(r_A, g) || Text1.$$

Конструювання маркера ключа (*B*1)

B випадково і таємно генерує r_B в *H*, обчислює $F(r_B, g)$ і надсилає до *A* маркер ключа

$$KT_{B1} = F(r_B, g) || Text2.$$

Конструювання ключа (*B*2)

B виділяє $F(r_A, g)$ з одержаного маркера ключа і обчислює розподілений таємний ключ як

$$K_{AB} = \omega(F(h_B,F(r_A,g)), F(r_B,p_A)),$$

де ω є односпрямованою функцією.

Конструювання ключа (*A*2)

A виділяє $F(r_B, g)$ з одержаного маркера ключа і обчислює розподілений таємний ключ як

$$K_{AB} = \omega(F(r_A, p_B), F(h_A, F(r_B, g))),$$

де ω є односпрямованою функцією.

Примітка. Цей механізм узгодження ключа має такі властивості:

1. Число проходів: 2.

2. Автентифікація ключа: цей механізм забезпечує взаємну неявну автентифікацію ключа. Якщо поле даних *Tetxt2* містить криптографічне контролювальне значення (на відомих даних), обчислене з використанням ключа K_{AB}, то механізм забезпечує явну автентифікацію ключа від *B* до *A*.

3. Підтвердження ключа: якщо поле даних *Tetxt2* містить криптографічне контролювальне значення (на відомих даних), обчислене з використанням ключа K_{AB}, то цей механізм забезпечує підтвердження ключа від *B* до *A*.

4. Це механізм узгодження ключа внаслідок того, що встановлений ключ є односпрямованою функцією випадкових чисел r_A і r_B, постачених, відповідно, *A* і *B*. Однак, унаслідок того, що суб'єкт *B* може знати $F(r_A, g)$ перед тим, як обрати значення r_B, суб'єкт *B* може обрати орієнтовно *s* бітів ключа, що встановлюється, за рахунок генерації 2^s пробних значень r_B за час від одержання $КТ_{A1}$ до надсилання $КТ_{B1}$.

5. *Приклад:* приклад цього механізму погодження ключа (відомий як схема погодження ключа Мацумото Такашіма-Імаі *A*(0)) наведений в В.6. Інший приклад відомий як протокол Госса.

6. Функція ω повинна приховувати свої вхідні дані в тому сенсі, що за значенням функції і одного з вхідних даних неможливо обчислити відповідну частину інших вхідних даних. Цього можна досягти, використовуючи ґеш-функцію з ISO/IEC 10118 (нема потреби в колізійно-стійкій ґеш-функції).

7. Сертифікати відкритих ключів: якщо *Tetxt1* і *Tetxt2* містять сертифікати відкритих ключів узгодження ключа, відповідно, суб'єктів *A* або *B*, вимога 2 на початку цього розділу може бути замінена вимогою, за якою кожен суб'єкт володіє автентифікованою копією відкритого ключа верифікації СА.

6.6 Узгодження ключа, механізм 6

Цей механізм узгодження ключа за два проходи встановлює розподілений таємний ключ між суб'єктами *A* і *B* з неявною взаємною автентифікацією ключа і спільним контролем ключа. Він оснований на використанні як асиметричного шифрування, так і системи підпису. Повинні задовольнятися такі вимоги:

1. *A* має асиметричну систему шифрування з перетворенням (E_A, D_A).

2. *B* має асиметричну систему підпису з перетворенням (S_B, V_B).

3. *A* має доступ до автентифікованої копії відкритого перетворення верифікації суб'єкта *B* — V_B. Цього можна досягти, використовуючи механізми розділу 8.

4. *B* має доступ до автентифікованої копії відкритого перетворення зашифровування E_A суб'єкта *A*. Цього можна досягти, використовуючи механізми розділу 8.

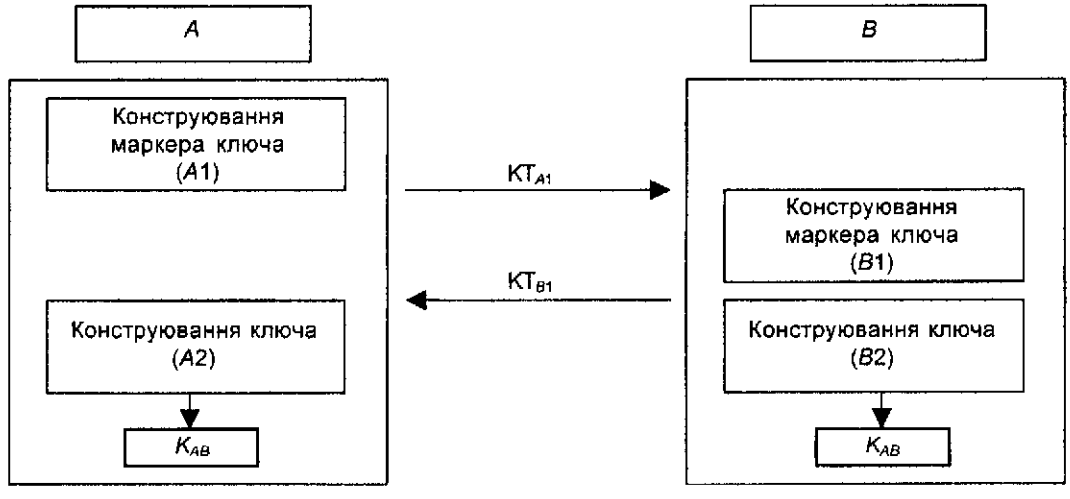

Рисунок 6 — Узгодження ключа, механізм 6

Конструювання маркера ключа (*A*1)

A генерує випадкове число r_A і надсилає до *B* маркер ключа

$$KT_{A1} = r_A || Text1.$$

Обробляння маркера ключа (*B*1)

B генерує випадкове число r_B і підписує блок даних, який складається з розрізнювального ідентифікатора *A*, випадкового числа r_A і деяких, необов'язкових, даних *Text2*, використовуючи своє приватне перетворення підписування S_B

$$BS = S_B(A || r_A || r_B || Text2).$$

Потім *B* зашифровує блок даних, який складається з його розрізнювального ідентифікатора *B* (необов'язково), підписаного блока BS і деяких, необов'язкових, даних *Text3*, використовуючи відкрите перетворення зашифровування E_A суб'єкта *A*, і надсилає назад до *A* маркер ключа

$$KT_{B1} = E_A(B || BS || Text3) || Text4.$$

Конструювання ключа (*B*2)

Таємний розподілений ключ складається з усього або частини підпису $B — \Sigma$, який міститься в підписаному блоці BS (див. примітку 1 розділу 4).

Обробляння маркера ключа (*A*2)

A розшифровує маркер ключа KT_{B1} із використанням свого приватного перетворення розшифровування D_A, необов'язково, контролює ідентифікатор посилача *B*, і використовує відкрите перетворення верифікації V_B суб'єкта *B* для верифікації цифрового підпису підписаного блока BS. Потім *A* контролює ідентифікатор одержувача *A* і узгодженість випадкового числа r_A в підписаному блоці BS із випадковим числом r_A, надісланим у маркері KT_{A1}. Якщо всі перевірки успішні, *A* приймає весь або частину підпису $B — \Sigma$ з підписаного блока BS як розподілений таємний ключ.

Примітка. Цей механізм узгодження ключа має такі властивості:

1. Число проходів: 2.

2. Автентифікація ключа: механізм забезпечує неявну автентифікацію ключа від *A* до *B* і явну автентифікацію ключа від *B* до *A*.

3. Підтвердження ключа: якщо поле даних *Text3* містить криптографічне контролювальне значення (на відомих даних), обчислене з використанням ключа K_{AB}, то цей механізм забезпечує підтвердження ключа від *B* до *A*.

4. Це механізм узгодження ключа внаслідок того, що встановлений ключ є односпрямованою функцією випадкових чисел r_A і r_B постачених, відповідно, *A* і *B*. Однак, внаслідок того, що суб'єкт *B* може знати $F(r_A, g)$, перед тим, як обрати значення r_B, суб'єкт *B* може обрати орієнтовно *s* бітів ключа, що встановлюється, за рахунок генерації 2^s пробних значень r_B за час від одержання KT_{A1} до надсилання KT_{B1}.

5. *Приклад*: цей механізм виведено з двопрохідного протоколу Беллера і Якобі, який наведено в розділі B.7.

6. Сертифікати відкритих ключів: якщо *Text1* і *Text4* містять сертифікат відкритого ключа зашифрування суб'єкта *A* і сертифікат відкритого ключа верифікації суб'єкта *B*, відповідно, то вимоги 3 і 4 на початку цього розділу можуть бути послаблені до вимоги, що кожен суб'єкт володіє автентифікованою копією відкритого ключа верифікації суб'єкта CA.

7. Важливою характеристикою цієї схеми є те, що тотожність сторони *B* може лишатись анонімною для підслуховувачів, визначною перевагою в безпровідному середовищі, яке є основним середовищем для застосування цієї схеми.

6.7 Узгодження ключа, механізм 7

Цей механізм, оснований на трипрохідному механізмі автентифікації ISO/IEC 9798-3, встановлює за три проходи розподілення таємного ключа між суб'єктами *A* і *B* із взаємною автентифікацією. Повинні задовольнятися такі вимоги:

1. Кожен суб'єкт *X* має асиметричну систему підпису (S_X, V_X).

2. Кожен суб'єкт має доступ до автентифікованої копії відкритого перетворення верифікації іншого суб'єкта. Це може бути досягнуто використанням механізмів розділу 8.

3. Кожен суб'єкт має спільну криптографічну контролювальну функцію *f*.

Конструювання маркера ключа (*A*1)

A випадково і таємно генерує r_A в *H*, обчислює $F(r_A, g)$, конструює і надсилає до *B* маркер ключа

$$KT_{A1} = F(r_A, g) || Text1.$$

Обробляння маркера ключа і конструювання ключа (*B*1)

B випадково і таємно генерує r_B у *H*, обчислює $F(r_B, g)$, обчислює розподілений таємний ключ як

$$K_{AB} = F(r_B, F(r_A, g)).$$

Конструює і підписує маркер ключа

$$KT_{B1} = S_B(DB_1) || f_{KAB}(DB_1) || Text3,$$

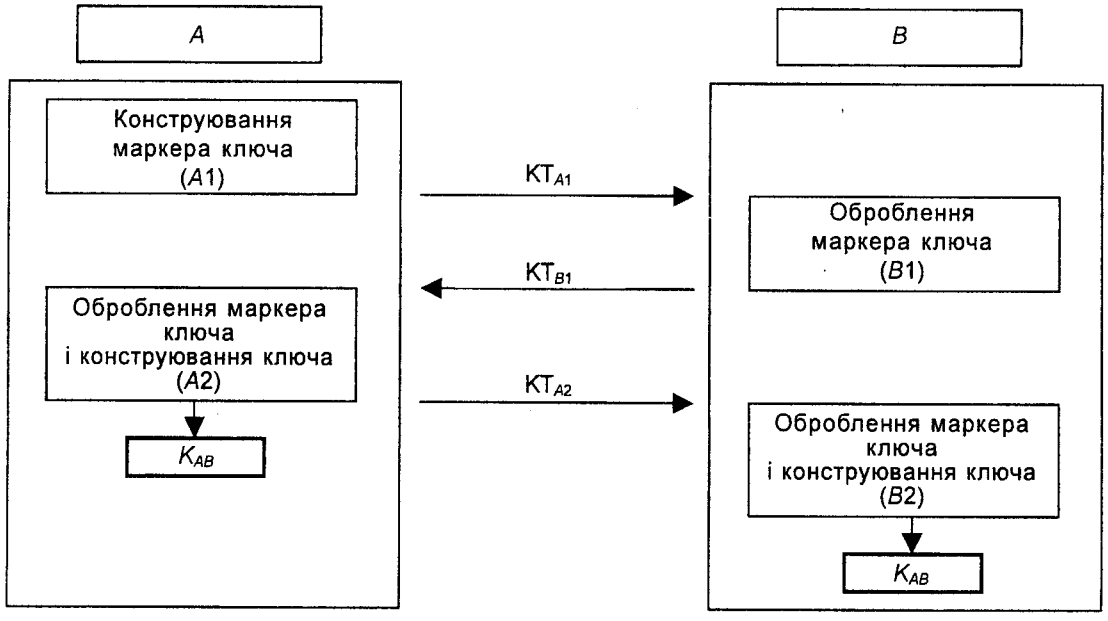

Рисунок 7 — Узгодження ключа, механізм 7

де

$$DB_1 = F(r_B, g)||F(r_A, g)||A||Text2$$

і надсилає його назад до *A*.

Підтвердження ключа забезпечують надсиланням $f_{KAB}(DB_1)$ у KT_{B1}. Альтернативно, якщо обидві сторони мають спільну симетричну систему шифрування, підтвердження ключа може бути отримано зашифровуванням частини маркера так: заміною KT_{B1} на $F(r_B, g)$, за яким слідує $E_{KAB}(S_B(DB_1))$.

Обробляння маркера ключа (*A*2)

A верифікує підпис суб'єкта *B* на маркері ключа KT_{B1} із використанням відкритого ключа верифікації *B*, верифікує розрізнювальний ідентифікатор суб'єкта *A* і значення $F(r_A, g)$, надіслане на кроці (*A*1). Якщо всі перевірки успішні, *A* продовжує обчислювання розподіленого таємного ключа як:

$$K_{AB} = F(r_A, F(r_B, g)).$$

Використовуючи K_{AB}, *A* перевіряє криптографічне контролювальне значення $f_{KAB}(DB_1)$.

Потім *A* конструює і надсилає до *B* підписаний маркер ключа

$$KT_{A2} = S_A(DB_2) \,||\, f_{KAB}(DB_2)||Text5,$$

де

$$DB_2 = F(r_A, g)||F(r_B, g)||B||Text4.$$

Підтвердження ключа забезпечують надсиланням $f_{KAB}(DB_2)$ в KT_{A2}. Альтернативно, підтвердження ключа може бути отримано зашифровуванням частини маркера таким чином: заміною KT_{A2} на $E_{KAB}(S_B(DB_2))$.

Обробляння маркера ключа (*B*2)

B верифікує підпис суб'єкта *A* на маркері ключа KT_{A2} з використанням відкритого ключа верифікації *A*, потім верифікує розрізнювальний ідентифікатор суб'єкта *B* і узгодженість значень $F(r_A, g)$ і $F(r_B, g)$ зі значеннями, якими обмінювалися на попередніх кроках. Якщо перевірки успішні, *B* верифікує криптографічне контролювальне значення $f_{KAB}(DB_2)$ з використанням

$$K_{AB} = F(r_B, F(r_A, g)).$$

Примітка. Цей механізм узгодження ключа має такі властивості:

1. Число проходів: 3.
2. Автентифікація ключа і суб'єкта: цей механізм забезпечує взаємну явну автентифікацію ключа і взаємну автентифікацію суб'єктів.
3. Підтвердження ключа: механізм забезпечує взаємне підтвердження ключа.

4. Це механізм узгодження ключа внаслідок того, що встановлений ключ є односпрямованою функцією випадкових чисел r_A і r_B постачених, відповідно, *A* і *B*. Однак, внаслідок того, що суб'єкт *B* може знати $F(r_A, g)$, перед тим, як обрати значення r_B, суб'єкт *B* може обрати орієнтовно *s* бітів ключа, що встановлюється, за рахунок генерації 2^s пробних значень r_B за час від одержання KT_{A1} до надсилання KT_{B1}.

5. *Приклад*: приклад цього механізму узгодження ключа може бути забезпечений за схемою Діффі-Гелмана, яку наведено в додатку В у поєднанні зі схемою цифрового підпису, наприклад, з ISO/IEC 9796.

6. Стандарти: цей механізм задовольняє ISO/IEC 9798-3 Автентифікація суб'єктів із використанням алгоритмів із відкритим ключем. KT_{A1}, KT_{B1} і KT_{A2} тотожні з маркерами, які надсилають у трипрохідному механізмі, наведеному в 5.2.2 ISO/IEC 9798-3. Також тотожні поля даних із такою заміною використання:

— випадкове значення функції $F(r_A, g)$ міститься в полі даних R_A (присутнє у всіх трьох маркерах ISO/IEC 9798-3, підрозділ 5.2.2)

— випадкове значення функції $F(r_B, g)$ міститься в полі даних R_B (присутнє у всіх трьох маркерах ISO/IEC 9798-3, підрозділ 5.2.2) .

7. Сертифікати відкритих ключів: якщо кожен з *Text1* і *Text3* (або *Text5* і *Text3)* містять сертифікат відкритого ключа суб'єктів *A* і *B*, відповідно, то вимога 2 на початку цього розділу може бути послаблена до вимоги, що кожен суб'єкт володіє автентифікованою копією відкритого ключа верифікації суб'єкта CA.

8. Перетворення підписування: якщо використовують механізм із ґешованим текстом, нема потреби надсилати $F(r_A, g)$ і (або) $F(r_B, g)$ в маркері ключа KT_{B1}. Аналогічно, нема потреби надсилати ні $F(r_A, g)$, ні $F(r_B, g)$, в маркері ключа KT_{A2}. Однак, необхідно потурбуватися, щоб випадкові номери були включені до обчислювання відповідних підписів.

7 ПЕРЕДАВАННЯ ТАЄМНОГО КЛЮЧА

У цій частині стандарту передавання ключа є процесом пересилання таємного ключа, обраного одним суб'єктом (або довірчим центром) іншому суб'єкту, відповідним чином захищеним асиметричним методом.

Примітка. У практичному запроваджені механізму передавання ключа блок даних ключа може підлягати додатковому оброблянню перед його використанням для зашифровування. Наприклад, блок даних ключа може бути спотвореним (псевдо-) випадковим бітовим шаблоном, щоб розрушити будь-яку очевидну структуру блока даних ключа.

7.1 Передавання ключа, механізм 1

Цей механізм передавання ключа пересилає за один прохід таємний ключ від суб'єкта *A* до суб'єкта *B* з неявною автентифікацією ключа від *B* до *A*. Повинні задовольнятися такі вимоги:

1. Суб'єкт *B* має асиметричну систему шифрування (E_B, D_B).

2. *A* має доступ до автентифікованої копії відкритого перетворення зашифрування E_B суб'єкта *B*. Цього можна досягти використанням механізмів розділу 8.

3. Необов'язковий *TVP* може бути або позначкою часу, або порядковим номером. Якщо використовують позначки часу, то суб'єкти *A* і *B* повинні вести синхронні годинники або використовувати третю довірчу сторону, повноважного позначальника часу. Якщо використовують порядкові номери, *A* і *B* повинні вести двосторонні лічильники.

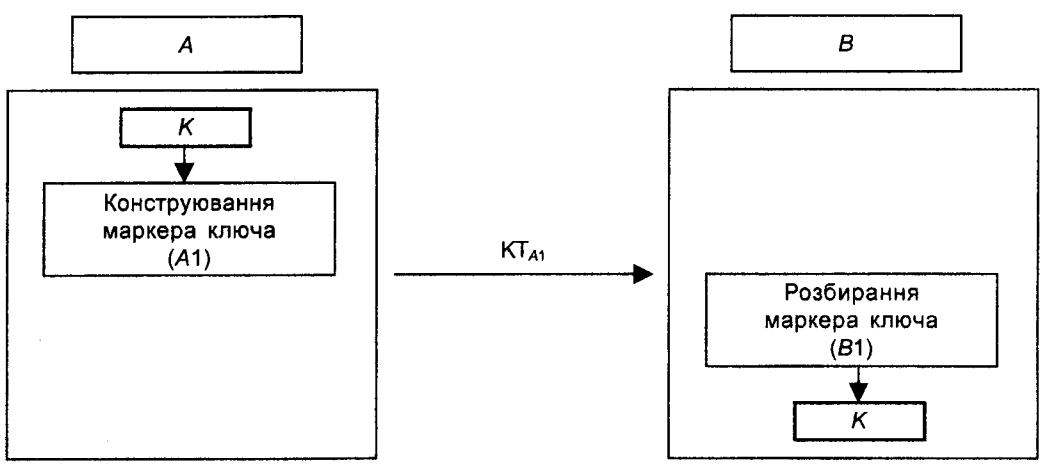

Рисунок 8 — Передавання ключа, механізм 1

Конструювання ключа (*A*1)

A отримав ключ *K* і хоче надійно передати його *B*. *A* конструює блок даних ключа, який складається з розрізнювального ідентифікатора *A* (необов'язково), ключа *K*, необов'язкового *TVP* і необов'язкового поля даних *Text1*. Потім *A* зашифровує блок даних ключа з використанням відкритого перетворення зашифрування одержувача E_B і надсилає до *B* маркер ключа

$$KT_{A1} = E_B(A||K||TVP||Text1)||\ Text2.$$

Розбирання маркера ключа (*B*1)

В розшифровує одержаний маркер ключа KT_{A1} із використанням свого приватного перетворення розшифровування D_B, відновлює ключ *K*, контролює необов'язковий *TVP* і пов'язує відновлений ключ *K* із заявленим автором *A*.

Примітка. Цей механізм передавання ключа має такі властивості:

1. Число проходів: 1.
2. Автентифікація ключа: цей механізм забезпечує неявну автентифікацію ключа від *B* до *A* внаслідок того, що лише *B* має можливість відновити ключ *K*.
3. Підтвердження ключа: цей механізм не забезпечує підтвердження ключа.
4. Керування ключем: *A* може вибирати ключ.
5. *TVP*: необов'язковий *TVP* запобігає повторенню маркера ключа.
6. Використання ключа: внаслідок того, що *B* одержує ключ *K* від неавтентифікованого суб'єкта *A*, безпечне використання *K* суб'єктом *B* обмежене функціями, які не вимагають довіри до автентичності *A*, такі як розшифровування і генерування коду автентифікації повідомлення.
7. *Приклад*: приклад цього механізму (відомий як пересилання ключа за ЕльГамалем) наведений в В.8. Інший приклад цього механізму з використанням RSA наведений в В.10.

7.2 Передавання ключа, механізм 2

Цей механізм передавання ключа є поширенням однопрохідного механізму автентифікації суб'єкта ISO/IEC 9798-3. Він пересилає зашифрований і підписаний таємний ключ від суб'єкта *A* до суб'єкта *B* із неявною автентифікацією ключа від *A* до *B*. Повинні задовольнятися такі вимоги:

1. Суб'єкт *A* має асиметричну систему підпису (S_A, V_A).
2. Суб'єкт *B* має асиметричну систему шифрування (E_B, D_B).
3. Суб'єкт *A* має доступ до автентифікованої копії відкритого перетворення зашифровування E_B суб'єкта *B*. Цього можна досягти використанням механізмів розділу 8.
4. Суб'єкт *B* має доступ до автентифікованої копії відкритого перетворення верифікації V_A суб'єкта *A*. Цього можна досягти використанням механізмів розділу 8.
5. Необов'язковий *TVP* повинен бути або позначкою часу, або порядковим номером. Якщо використовують позначки часу, то суб'єкти *A* і *B* повинні вести синхронні годинники або використовувати третю довірчу сторону, повноважного позначальника часу. Якщо використовують порядкові номери, *A* і *B* повинні вести двосторонні лічильники.

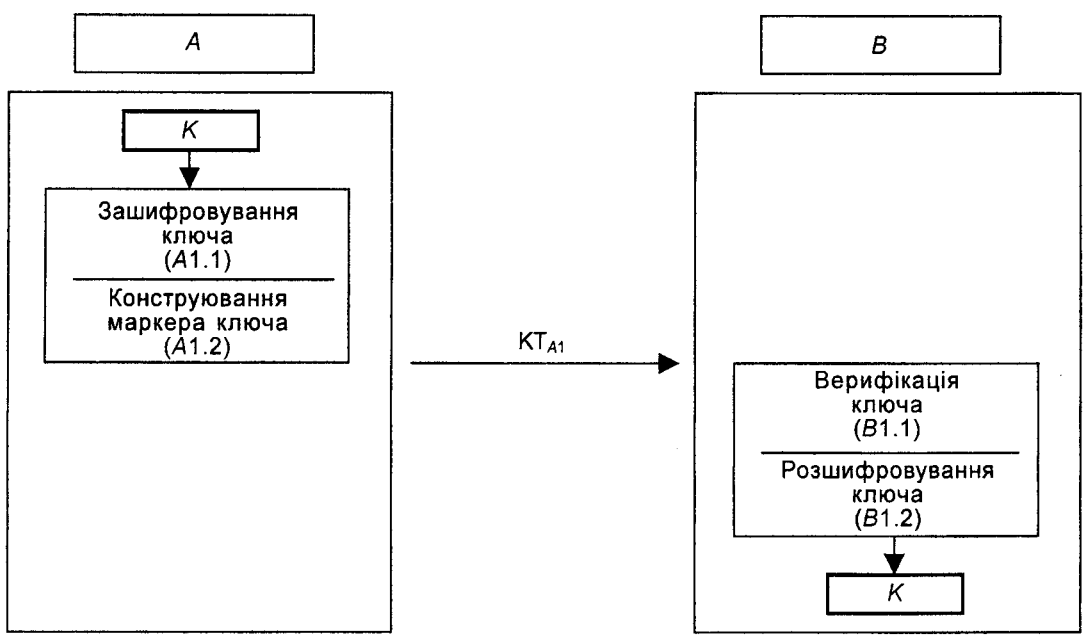

Рисунок 9 — Передавання ключа, механізм 2

Зашифровування ключа (*A*1.1)

A отримав ключ *K* і хоче надійно переслати його *B*. *A* формує блок даних ключа, який складається з розрізнювального ідентифікатора *A*, ключа *K*, і необов'язкового поля даних *Text1*. Потім *A* зашифровує блок даних ключа відкритим перетворенням зашифровування E_B суб'єкта *B* і формує зашифрований блок

$$\text{BE} = E_B(A||K||Text1).$$

Конструювання маркера ключа (*A*1.2)

A формує блок даних маркера, який містить розрізнювальний ідентифікатор одержувача *B*, необов'язкову позначку часу або порядковий номер *TVP*, зашифрований блок BE і необов'язкове поле даних *Text2*. Потім *A* підписує блок даних маркера з використанням свого приватного перетворення підписування S_A і надсилає до *B* остаточний маркер ключа

$$\text{KT}_{A1} = S_A(B||TVP||\text{BE}||Text2)||\ Text3.$$

Верифікація маркера ключа (*B*1.1)

B використовує відкрите перетворення верифікації посилача V_A, щоб верифікувати цифровий підпис одержаного маркера ключа KT_{A1}. Потім *B* контролює одержану ідентифікацію *B* і, необов'язково, *TVP*.

Розшифровування ключа (*B*1.2)

B розшифровує блок BE з використанням свого приватного перетворення розшифровування D_B. Потім *B* порівнює поле *A* в блоці BE з тотожністю суб'єкта, який підписав. Якщо всі перевірки успішні, *B* приймає ключ *K*.

Примітка. Цей механізм передавання ключа має такі властивості:

1. Число проходів: 1.

2. Автентифікація ключа і суб'єкта: цей механізм забезпечує автентифікацію суб'єкта *A* для *B*, якщо використовують необов'язковий *TVP*, та неявну автентифікацію ключа від *B* до *A*.

3. Підтвердження ключа: від *A* до *B*. *B* може бути впевнений, що він розподіляє з *A* правильний ключ, але *A* може бути впевнений в тому, що *B* дійсно одержав ключ, лише після того, як він отримає позитивну відповідь від *B*.

4. Керування ключем: *A* може обирати ключ.

5. (Необов'язковий) *TVP*: забезпечує автентифікацію суб'єкта *A* для *B* і запобігає повторенню маркера ключа. З метою запобігання повторенню блока даних ключа BE, в *Text1* може також розміщуватися додатковий *TVP*.

6. Поле даних *A*: розрізнювальний ідентифікатор *A* включається до зашифрованого блока BE для убезпечення *A* від помилкового прийняття зашифрованого блока ключа, який має намір використати інший суб'єкт. Це досягається порівнянням тотожності *A* з підписом *A* на маркері.

7. Стандарти: задовольняє ISO/IEC 9798-3 Автентифікація суб'єктів із використанням алгоритмів відкритих ключів. KT_{A1} сумісний з маркером, який надсилають в однопрохідному механізмі автентифікації, наведеному в 5.1.1 ISO/IEC 9798-3. Маркер пристосовує пересилання ключа *K* використанням необов'язкових текстових полів: *Text1* повинен бути замінений на BE||*Text2*.

8. Сертифікати відкритих ключів: поле даних *Text3* може бути використане для доставляння сертифіката відкритого ключа суб'єкта *A*. У такому разі, вимога 4 на початку цього розділу може бути послаблена до вимоги, що *B* володіє автентифікованою копією відкритого ключа верифікації CA.

9. Взаємна автентифікація суб'єктів і сумісний контроль ключів: якщо комбінують два виконання цього механізму пересилання ключа (від *A* до *B* і від *B* до *A*), то можуть бути забезпечені взаємна автентифікація суб'єктів і сумісний контроль ключів (залежно від використання необов'язкового *TVP*).

10. Використання: механізм пересилання ключа 2 спрямований на використання в середовищі, де вимагається конфіденційність сторін під час обміну повідомленнями, тобто повідомлення, яке несе багато неконфіденційних елементів, а також зашифрованих ключів.

11. Приклади цього механізму наведено в B.9 і C.7.

7.3 Передавання ключа, механізм 3

Цей механізм пересилає за один прохід таємний ключ, підписаний і зашифрований суб'єктом *A*, суб'єкту *B* з одностороннім підтвердженням ключа. Повинні задовольнятися такі вимоги:

1. Суб'єкт *A* має асиметричну систему підпису (S_A, V_A).

2. Суб'єкт *B* має асиметричну систему шифрування (E_B, D_B).

3. Суб'єкт *A* має доступ до автентифікованої копії відкритого перетворення зашифровування E_B суб'єкта *B*. Цього можна досягти використанням механізмів розділу 8.

4. Суб'єкт *B* має доступ до автентифікованої копії відкритого перетворення верифікації V_A суб'єкта *A*. Цього можна досягти використанням механізмів розділу 8.

5. Необов'язковий *TVP* повинен бути або позначкою часу, або порядковим номером. Якщо використовують позначки часу, то суб'єкти *A* і *B* повинні вести синхронні годинники. Якщо використовують порядкові номери, *A* і *B* повинні вести двосторонні лічильники.

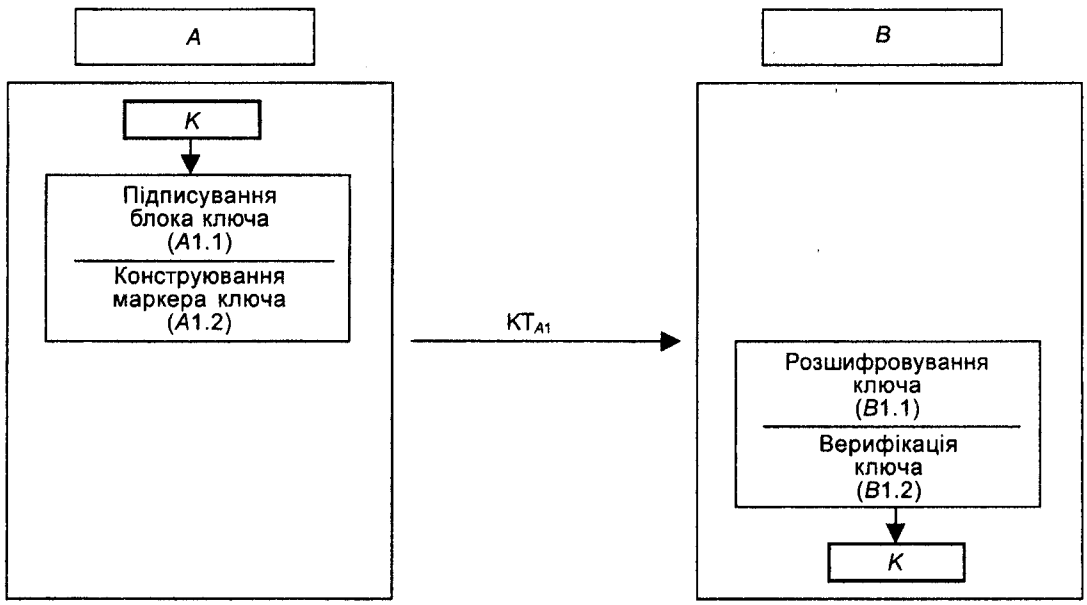

Рисунок 10 — Передавання ключа, механізм 3

Підписування блока ключа (*A*.1.1)

A отримав ключ *K* і хоче надійно переслати його *B*. *A* формує блок даних ключа, який складається з розрізнювального ідентифікатора одержувача *B*, ключа *K*, і необов'язкової позначки часу, або порядкового номера *TVP*. Потім *A* підписує блок даних ключа з використанням свого приватного перетворення підписування S_A для генерування підписаного блока

$$BS = S_A(B||K||TVP||Text1).$$

Конструювання маркера ключа (*A*1.2)

A формує блок даних маркера, який складається з підписаного блока BS і деякого необов'язкового *Text2*. Потім *A* зашифровує блок даних маркера з використанням відкритого перетворення зашифровування одержувача E_B і надсилає до *B* остаточний маркер ключа

$$KT_{A1} = E_B(BS||Text2)||Text3.$$

Розшифровування маркера ключа (*B*1.1)

B розшифровує одержаний маркер ключа KT_{A1} із використанням свого приватного перетворення розшифровування D_B.

Верифікація блока ключа (*B*1.2)

B використовує відкрите перетворення верифікації посилача V_A для верифікації цілісності і походження BS. *B* підтверджує, що це передбачуваний одержувач маркера (оглядом ідентифікатора *B*) і, необов'язково, щоб маркер був надісланий своєчасно (оглядом *TVP*). Якщо всі верифікації успішні, *B* приймає ключ *K*.

Примітка. Цей механізм передавання ключа має такі властивості:

1. Число проходів: 1.

2. Автентифікація ключа і суб'єкта: цей механізм забезпечує автентифікацію суб'єкта *A* для *B*, якщо використовують необов'язковий *TVP*, та неявну автентифікацію ключа від *B* до *A*.

3. Підтвердження ключа: від *A* до *B*. *B* може бути впевнений, що він розподіляє з *A* правильний ключ, але *A* може бути впевнений у тому, що *B* дійсно одержав ключ лише після того, як він отримає позитивну відповідь від *B*.

4. Керування ключем: *A* може обирати ключ.

5. (Необов'язковий) *TVP*: може забезпечувати автентифікацію суб'єкта *A* для *B* і запобігати повторенню маркера ключа.

6. Поле даних *B*: розрізнювальний ідентифікатор *B* міститься в зашифрованому блоці BS для явного зазначення одержувача ключа, таким чином запобігаючи зловживанню зі сторони *B* блоком BS.

7. Сертифікати відкритих ключів: поле даних *Text3* може бути використане для доставляння сертифіката відкритого ключа суб'єкта *A*. У такому разі, вимога 3 на початку цього розділу може бути послаблена до вимоги, що *B* володіє автентифікованою копією відкритого ключа верифікації CA.

8. Взаємна автентифікація суб'єктів і сумісний контроль ключів: якщо комбінують два виконання цього механізму пересилання ключа (від *A* до *B* і від *B* до *A*), то можуть бути забезпечені взаємна автентифікація суб'єктів і сумісний контроль ключів (залежно від використання необов'язкового *TVP*).

7.4 Передавання ключа, механізм 4

Цей механізм передавання ключа, оснований на двопрохідному механізмі автентифікації ISO/IEC 9798-3, пересилає ключ від суб'єкта *B* до суб'єкта *A*. Повинні задовольнятися такі вимоги:

1. Суб'єкт *A* має асиметричну систему шифрування (E_A, D_A).
2. Суб'єкт *B* має асиметричну систему підпису (S_B, V_B).
3. Суб'єкт *A* має доступ до автентифікованої копії відкритого перетворення верифікації V_B суб'єкта *B*. Цього можна досягти використанням механізмів розділу 8.
4. Суб'єкт *B* має доступ до автентифікованої копії відкритого перетворення зашифровування E_A суб'єкта *A*. Цього можна досягти використанням механізмів розділу 8.

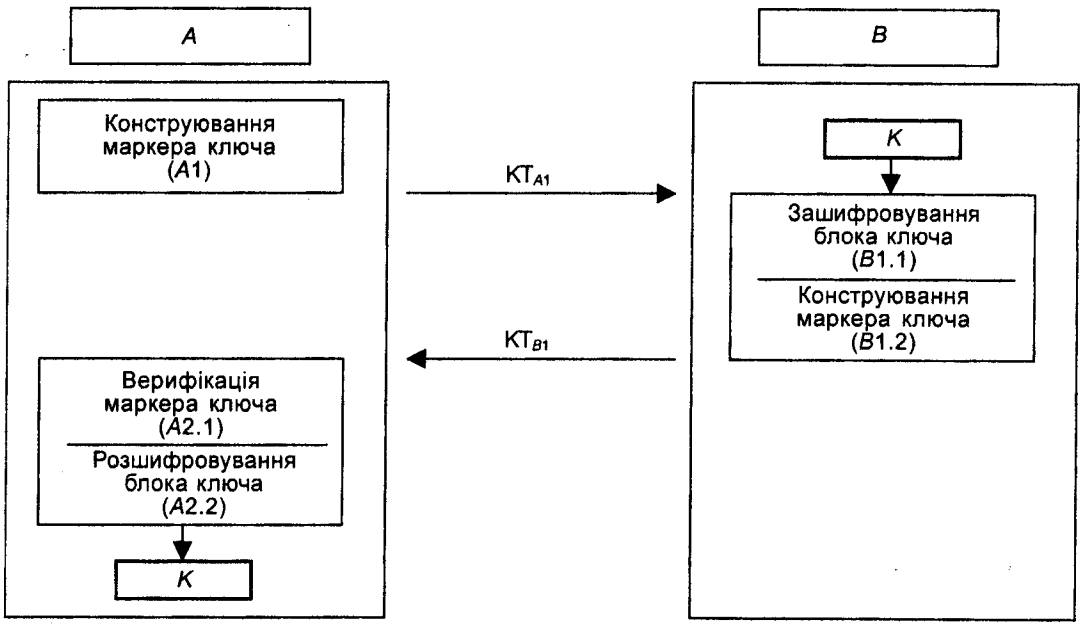

Рисунок 11 — Передавання ключа, механізм 4

Конструювання маркера ключа (*A*1)

A конструює маркер ключа KT_{A1}, який складається з випадкового числа r_A і необов'язкового поля даних *Text1*

$$KT_{A1} = r_A || Text1$$

і надсилає його до *B*.

Зашифровування блока ключа (*B*1.1)

B отримав ключ *K* і хоче надійно переслати його *A*. *B* формує блок даних ключа, який складається з розрізнювального ідентифікатора посилача *B*, ключа *K*, і необов'язкового поля даних *Text2*. Потім *B* зашифровує блок даних ключа відкритим перетворенням зашифровування E_A суб'єкта *A* і формує зашифрований блок

$$BS = E_A(B||K||Text2).$$

Конструювання маркера ключа (*B*1.2)

B формує блок даних маркера, який складається з розрізнювального ідентифікатора одержувача *A*, випадкового числа r_A, одержаного на кроці (*A*1), нового випадкового числа r_B, необов'язково, зашифрованого блока BE і необов'язкового поля даних *Text3*. Потім *B* підписує блок даних маркера своїм приватним перетворенням підпису S_B і надсилає до *A* остаточний маркер ключа

$$KT_{B1} = S_B(A||r_A||r_B||BE||Text3)||Text4.$$

Верифікація маркера ключа (*A*2.1)

A використовує відкрите перетворення верифікації посилача V_B для верифікації цифрового підпису одержаного маркера ключа KT_{B1}. Потім *A* контролює розрізнювальний ідентифікатор *A*

і контролює узгодження одержаного випадкового числа r_A з випадковим числом, надісланим на кроці (*A*1).

Розшифровування блока ключа (*A*2.2)

A розшифровує блок ВЕ своїм приватним перетворенням розшифровування D_A. Потім *A* підтверджує розрізнювальний ідентифікатор посилача *B*. Якщо всі перевірки успішні, *A* приймає ключ *K*.

Примітка. Цей механізм передавання ключа має такі властивості:

1. Число проходів: 2.

2. Автентифікація ключа і суб'єкта: цей механізм забезпечує автентифікацію суб'єкта *B* для *A* та неявну автентифікацію ключа від *A* до *B*.

3. Підтвердження ключа: від *B* до *A*. *A* може бути впевнений, що він роздподілює з *B* правильний ключ *K*, але *B* може бути впевнений у тому, що *A* дійсно одержав ключ лише після того, як він отримає захищене повідомлення від *A*, яке повинно бути недвозначно оброблено.

4. Керування ключем: *B* може обирати ключ.

5. Стандарти: задовольняє ISO/IEC 9798-3. Маркери $КТ_{A1}$ і $КТ_{B1}$ сумісні з маркерами, які надсилаються в двопрохідному механізмі автентифікації, наведеному в розділі 5.1.2 ISO/IEC 9798-3 (зазначимо, що ролі *A* і *B* переставлені). Маркер $КТ_{B1}$ пристосовує пересилання ключа *K* використанням необов'язкових полів даних: *Text2* повинне бути замінене на *Text3*.

6. Стандарти: якщо механізм пересилання ключа виконується двічі паралельно між двома суб'єктами, то остаточний взаємний механізм пересилання ключа задовольняє механізм, наведений в 5.2.3 ISO/IEC 9798-3.

7. Поле даних r_B: вказано для сумісності з 9798-3. Унаслідок наявності ВЕ в $КТ_{B1}$, поле даних r_B більше не потрібне, і тому є необов'язковим у цьому механізмі.

8. Взаємна автентифікація суб'єктів і сумісний контроль ключів: якщо комбінують два виконання цього механізму пересилання ключа (від *A* до *B* і від *B* до *A*), то можуть бути забезпечені взаємна автентифікація суб'єктів і сумісний контроль ключів.

7.5 Передавання ключа, механізм 5

Цей механізм передавання ключа, оснований на трипрохідному механізмі автентифікації ISO/IEC 9798-3, пересилає за три проходи два розподілених таємних ключа із взаємною автентифікацією суб'єктів і підтвердженням ключів. Один ключ передають від *A* до *B* і один ключ від *B* до *A*. Повинні задовольнятися такі вимоги:

1. Кожен суб'єкт *X* має асиметричну систему підпису (S_X, V_X).
2. Кожен суб'єкт *X* має асиметричну систему шифрування (E_X, D_X).
3. Кожен суб'єкт *X* має доступ до автентифікованої копії відкритого перетворення верифікації іншого суб'єкта. Цього можна досягти використанням механізмів розділу 8.
4. Кожен суб'єкт *X* має доступ до автентифікованої копії відкритого перетворення зашифровування іншого суб'єкта. Цього можна досягти використанням механізмів розділу 8.

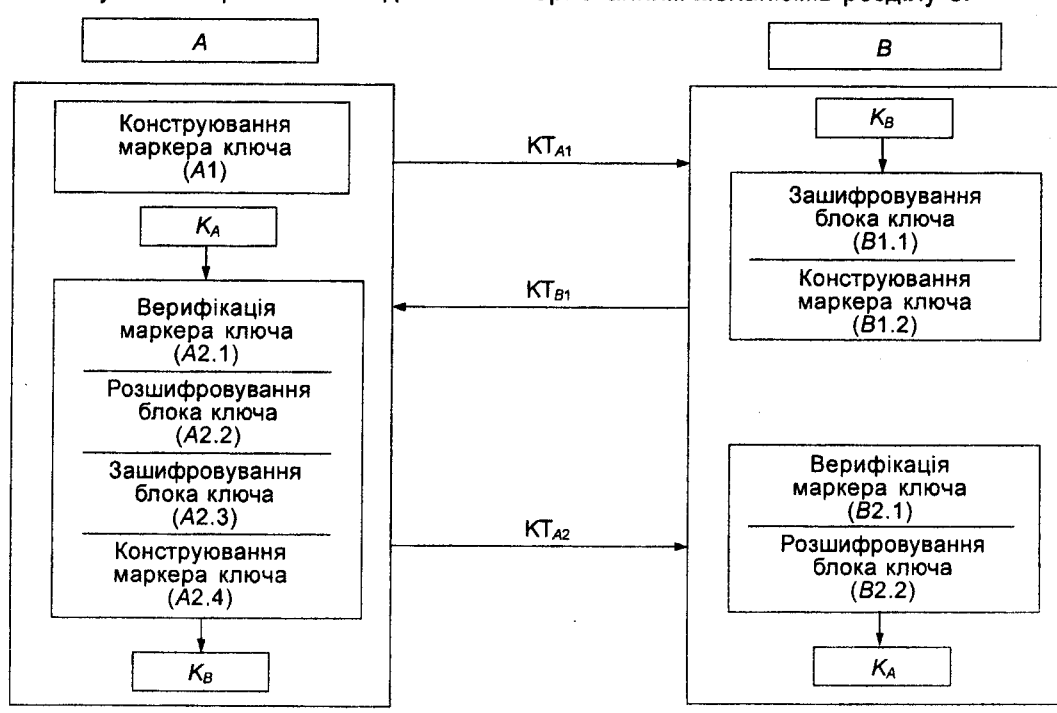

Рисунок 12 — Передавання ключа, механізм 5

Конструювання маркера ключа (*A*1)

A випадково генерує r_A, конструює маркер ключа

$$KT_{A1} = r_A || Text1$$

і надсилає його до *B*.

Зашифровування блока ключа (*B*1.1)

B отримав ключ *K* і хоче переслати його надійно до *A*. *B* конструює блок, який містить його власний розрізнювальний ідентифікатор *B*, ключ *K* і необов'язкове поле даних *Text2*, і зашифровує блок із використанням відкритого перетворення зашифровування E_A одержувача

$$BE_1 = E_A(B||K_B||Text2).$$

Конструювання маркера ключа (*B*1.2)

B випадково генерує r_B і конструює блок даних, який містить r_B, r_A, тотожність одержувача *A*, зашифрований блок BE_1, і деяке необов'язкове поле даних *Text3*. *B* підписує блок із використанням свого приватного перетворення підписування S_B і надсилає до *A* маркер ключа

$$KT_{B1} = S_B(r_B||r_A||\ A||\ BE_1||Text3)||Text4.$$

Верифікація маркера ключа (*A*2.1)

A верифікує підпис суб'єкта *B* на маркері ключа KT_{B1} з використанням відкритого перетворення верифікації V_B суб'єкта *B*, контролює розрізнювальний ідентифікатор *A* і контролює узгодженість одержаного значення r_A з випадковим числом, надісланим на кроці (*A*1).

Розшифровування блока ключа (*A*2.2)

A розшифровує зашифрований блок BE_1 із використанням свого приватного перетворення розшифровування D_A і контролює розрізнювальний ідентифікатор *B*. Якщо всі перевірки успішні, *A* приймає ключ K_B.

Зашифровування блока ключа (*A*2.3)

Потім *A* конструює блок даних, який містить його власний розрізнювальний ідентифікатор *A*, його власний ключ K_A і деяке необов'язкове поле даних *Text5*, та зашифровує блок із використанням відкритого перетворення зашифровування E_B

$$BE_2 = E_B(A||K_A||Text5).$$

Конструювання маркера ключа (*A*2.4)

Потім *A* конструює блок даних, який містить випадкове число r_A, випадкове число r_B, розрізнювальний ідентифікатор одержувача *B*, зашифрований блок ключа BE_2 і деяке необов'язкове *Text6*. *A* підписує блок даних із використанням свого приватного перетворення підпису S_A і надсилає до *B* маркер ключа

$$KT_{A2} = S_A(r_A||r_B||B||BE_2||Text6)||Text7.$$

Верифікація маркера ключа (*B*2.1)

B верифікує підпис суб'єкта *A* на маркері ключа KT_{A2} з використанням його відкритого перетворення верифікації V_A, контролює розрізнювальний ідентифікатор *B* і контролює узгодженість одержаного значення r_B із випадковим числом, надісланим на кроці (*B*1.2). Крім того, *B* контролює, що одержане значення r_A узгоджується з розміщеним у KT_{A1}.

Розшифровування блока ключа (*B*2.2)

B розшифровує зашифрований блок BE_2 з використанням свого приватного перетворення розшифровування D_B і контролює розрізнювальний ідентифікатор *A*. Якщо всі перевірки успішні, *B* приймає ключ K_A.

Якщо потрібне лише одностороннє передавання, то відповідно BE_1 або BE_2 може бути опущений.

Примітка. Цей механізм передавання ключа має такі властивості:

1. Число проходів: 3.

2. Автентифікація ключа і суб'єкта: цей механізм забезпечує взаємну автентифікацію суб'єктів, неявну автентифікацію ключа K_A від *B* до *A* і неявну автентифікацію ключа K_B від *A* до *B*.

3. Підтвердження ключа: цей механізм забезпечує підтвердження ключа від посилача до одержувача для обох ключів K_A і K_B. Більше того, якщо *A* містить криптографічне контролювальне значення для K_B у полі даних *Text6* в KT_{A2}, то механізм забезпечує взаємне підтвердження ключа по відношенню до K_B.

4. Керування ключем: *A* може обрати ключ K_A, бо він є суб'єктом-автором. Аналогічно *B* може обрати ключ K_B. Сумісне керування ключем може бути досягнуте кожним суб'єктом комбінуванням двох ключів K_A і K_B на обох сторонах для формування розподіленого таємного ключа K_{AB}. Однак, комбінована функція повинна бути односпрямованою, інакше *A* може обрати розподілений таємний ключ. Таким чином, цей механізм може бути класифікований як механізм узгодження ключа.

5. Стандарти: задовольняє ISO/IEC 9798-3. Маркери KT_{A1}, KT_{B1} і KT_{A2} сумісні з маркерами, які надсилають у трипрохідному механізмі автентифікації, наведеному в 5.2.2 ISO/IEC 9798-3. Другий маркер KT_{B1} пристосовує пересилання ключа K_B: *Text2* повинен бути замінений на $BE_1||Text3$. Третій маркер пристосовує пересилання ключа K_A: *Text4* повинен бути замінений на $BE_2||Text6$. Третій маркер може також пристосувати пересилання криптографічного контролювального значення в *Text6*.

6. Сертифікати відкритих ключів: якщо кожне із полів даних *Text1* і *Text4* (або *Text7* і *Text4*) містять сертифікати відкритих ключів суб'єктів *A* і *B*, відповідно, то вимоги 3 і 4 на початку цього розділу можуть бути послаблені до вимоги, щоб усі суб'єкти володіли автентифікованою копією відкритого ключа верифікації суб'єкта CA.

7. Перетворення підпису: якщо використовують механізм підписування з ґешуванням текста, то, необов'язково, випадкове число r_A не повинне надсилатись у маркері ключа KT_{B1}. Аналогічно ні r_A, ні r_B не повинні надсилатися в маркері ключа KT_{A2}. Однак, треба потурбуватися, щоб випадкові числа були включені в обчислення відповідних підписів.

7.6 Передавання ключа, механізм 6

Цей механізм передавання ключа пересилає за три проходи два таємних ключа: один від *A* до *B*, другий від *B* до *A*. Крім того, механізм забезпечує взаємну автентифікацію суб'єктів і взаємне підтвердження їхніх відповідних ключів. Цей механізм базований на таких вимогах:

1. Кожен суб'єкт *X* має асиметричну систему шифрування (E_X, D_X).
2. Кожен суб'єкт має доступ до автентифікованої копії відкритого перетворення зашифровування іншого суб'єкта. Цього можна досягти використанням механізмів розділу 8.

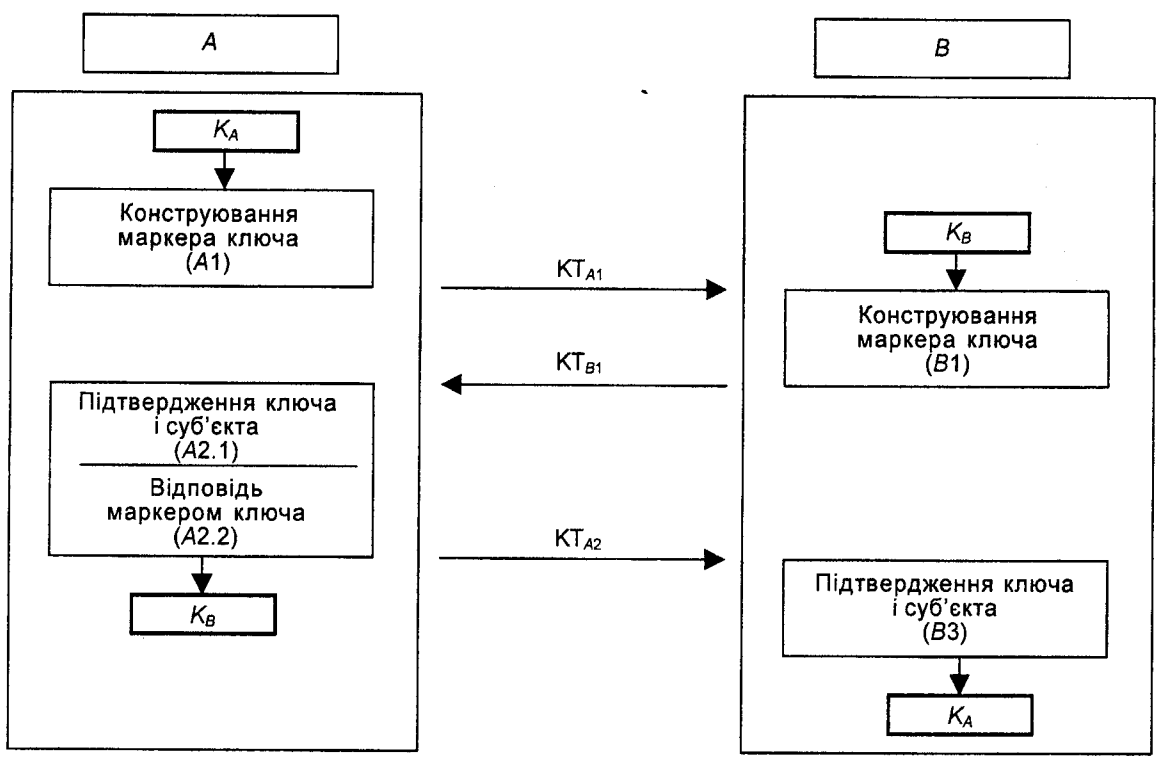

Рисунок 13 — Передавання ключа, механізм 6

Конструювання маркера ключа (A1)

A отримав ключ K_A і хоче його надійно переслати до *B*. *A* обирає випадкове число r_A і конструює блок даних ключа, який складається з його розрізнювального ідентифікатора *A*, ключа K_A числа r_A і необов'язкового поля даних *Text1*. Потім *A* зашифровує блок ключа з використанням відкритого перетворення зашифровування E_B суб'єкта *B*, таким чином виробляючи зашифрований блок даних

$$BE_1 = E_B(A \,||\, K_A \,||\, r_A \,||\, Text1).$$

A конструює маркер KT_{A1}, який складається з зашифрованого блока даних і деякого необов'язкового поля даних *Text2*

$$KT_{A1} = BE_1||Text2.$$

A надсилає маркер до *B*.

Конструювання маркера ключа (*B*1)

B виділяє зашифрований блок ключа BE_1 з одержаного маркера ключа KT_{A1} і розшифровує його з використанням свого приватного перетворення розшифрування D_B. Потім *B* перевіряє тотожність посилача *A*.

B отримав ключ K_A і хоче його надійно переслати до *A*. *B* обирає випадкове число r_B і конструює блок даних ключа, який складається з розрізнювального ідентифікатора *B*, ключа K_B, випадкового числа r_B, випадкового числа r_A (яке було виділене із зашифрованого блока) і необов'язкового поля даних *Text3*. Потім *B* зашифровує блок ключа з використанням відкритого перетворення зашифровування E_A суб'єкта *A*, таким чином виробляючи зашифрований блок даних

$$BE_2 = E_A(B \,||\, K_B \,||\, r_A \,||\, r_B \,||\, Text3).$$

Потім *B* конструює маркер ключа KT_{B1}, який містить зашифрований блок даних BE_2 і деяке необов'язкове поле даних *Text4*,

$$KT_{B1} = BE_2 \,||\, Text4.$$

B надсилає маркер до *A*.

Підтвердження ключа і суб'єкта (*A*2.1)

A виділяє зашифрований блок ключа BE_2 з одержаного маркера ключа KT_{B1} і розшифровує його з використанням свого приватного перетворення розшифровування D_A. Потім *A* контролює достовірність маркера ключа, порівнянням випадкового числа r_A з випадковим числом r_A, яке містилось у зашифрованому блоці BE_2. Якщо верифікація успішна, *A* автентифікував *B* і в той же час отримав підтвердження, що K_A успішно досяг суб'єкта *B*.

Відповідь маркером ключа (*A*2.2)

A виділяє випадкове число r_B із зашифрованого блока ключа і конструює маркер ключа KT_{A2}, який складається з випадкового числа r_B і необов'язкового поля даних *Text5*

$$KT_{A2} = r_B || Text5.$$

A надсилає маркер до *B*.

Підтвердження ключа і суб'єкта (*B*2)

B верифікує, що відповідь r_B, виділене з зашифрованого блока KT_{A2}, узгоджується з випадковим числом r_B, надісланим у зашифрованій формі в KT_{A2}. Якщо верифікація успішна, *B* автентифікував *A* і в той же час отримав підтвердження, що K_B досягнув суб'єкта *A* в цілості.

Примітка. Цей механізм передавання ключа має такі властивості:

1. Число проходів: 3.

2. Автентифікація суб'єкта: цей механізм забезпечує взаємну автентифікацію суб'єктів, неявну автентифікацію ключа K_A від *B* до *A* і неявну автентифікацію ключа K_B від *A* до *B*.

3. Підтвердження ключа: цей механізм забезпечує взаємне підтвердження ключа.

4. Керування ключем: *A* може обрати ключ K_A, бо він є суб'єктом-автором. Аналогічно *B* може обрати ключ K_B. Сумісне керування ключем може бути досягнуте кожним суб'єктом комбінуванням двох ключів K_A і K_B на обох сторонах для формування розподіленого таємного ключа K_{AB}. Однак, комбінована функція повинна бути односпрямованою, інакше *B* може обрати розподілений таємний ключ. Таким чином, цей механізм може бути класифікований як механізм узгодження ключа.

5. Використання ключа: цей механізм використовує асиметричний метод для взаємного пересилання двох таємних ключів K_A від *A* до *B* і K_B від *B* до *A*. Наступне відокремлення криптографічної функції може бути виведене з механізму: *A* використовує свій ключ K_A для зашифровування повідомлень для *B* і верифікації кодів автентифікації від *B*. *B*, у свою чергу, використовує одержаний ключ K_A для розшифровування повідомлень від *A* і генерування кодів автентифікації для *A*. Криптографічні функції K_B можуть бути відокремленні аналогічним чином. Таким чином асиметрична основа механізму передавання ключа може бути поширена на використання таємних ключів.

6. *Приклад:* цей механізм виведений з трипрохідного протоколу, відомого як COMSET (див. Бібліографію, автор Brendt тощо).

7. Підґрунтя: цей механізм оснований на методі нульового знання. З виконання механізму жоден із суб'єктів не дізнається нічого, що він не міг би обчислити сам.

8 ПЕРЕДАВАННЯ ВІДКРИТОГО КЛЮЧА

У цьому розділі описано механізми керування ключами, які роблять відкриті ключі суб'єктів доступними для інших суб'єктів з їх автентифікацією. Автентифіковане розподілення відкритих ключів є суттєвою вимогою безпеки. Це автентифіковане розподілення може бути досягнуто різними шляхами:

1. Розподіленням відкритих ключів без третьої довірчої сторони;

2. Розподіленням відкритих ключів із залученням третьої довірчої сторони, такої як повноважний сертифікатор.

Відкритий ключ суб'єкта *A* є частиною інформації щодо відкритого ключа *A*. Інформація щодо відкритого ключа містить щонайменше розрізнювальний ідентифікатор *A* і відкритий ключ *A*.

8.1 Розподілення відкритих ключів без третьої довірчої сторони

У цьому розділі описано механізми, які забезпечують автентифіковане розподілення відкритих ключів без залучення третьої довірчої сторони.

8.1.1 *Розподілення відкритих ключів, механізм 1*

Якщо *A* має доступ до захищеного каналу до *B* (тобто каналу, який забезпечує автентифікацію походження даних і цілісність даних) такого, як кур'єр, реєстровна пошта тощо, то *A* може безпосередньо передати свою інформацію щодо відкритого ключа захищеним каналом до *B*. Це найпростіша форма передавання відкритого ключа. Повинні задовольнятися такі вимоги:

1. Інформація щодо відкритого ключа *A* — $РКІ_A$ — містить щонайменше розрізнювальний ідентифікатор суб'єкта *A* і його відкритий ключ. Додатково вона може містити номер серії, період дійсності, позначку часу і інші елементи даних.

2. У зв'язку з тим, що інформація щодо відкритого ключа РКІ не містить жодних таємних даних, канал взаємодії не повинен забезпечувати конфіденційність.

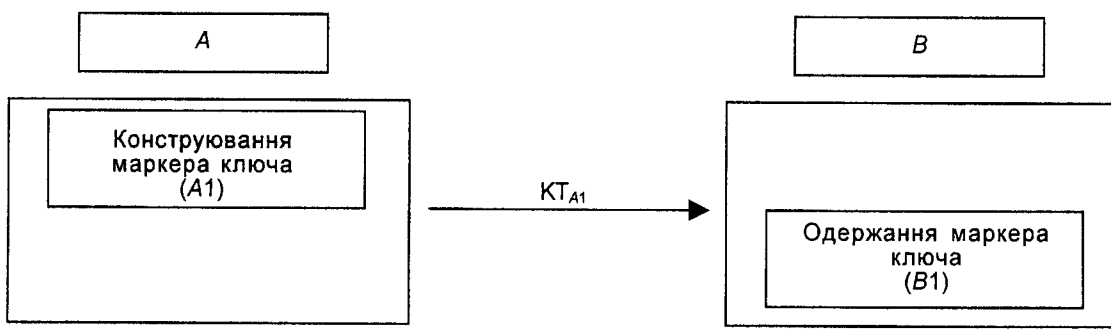

Рисунок 14 — Передавання відкритого ключа, механізм 1

Конструювання маркера ключа (*A*1)

A конструює маркер ключа $КТ_{A1}$, який містить інформацію щодо відкритого ключа суб'єкта *A* і деяке необов'язкове поле даних *Text*, і надсилає його до *B* захищеним каналом.

$$КТ_{A1} = РКІ_A \,||\, Text$$

Одержання маркера ключа (*B*1)

B одержує від *A* маркер ключа захищеним каналом, відновлює $РКІ_A$ — інформацію щодо відкритого ключа суб'єкта *A* і запам'ятовує відкритий ключ *A* в переліку активних відкритих ключів (цей перелік повинен бути захищеним від втручання).

Примітка. Цей механізм передавання відкритого ключа має такі властивості:

1. Цей механізм можна використовувати для пересилання відкритого ключа верифікації (для асиметричних систем підпису) або відкритого ключа зашифровування (для асиметричних систем шифрування), або відкритого ключа узгодження ключів.

2. Автентифікація в цьому контексті включає автентифікацію як цілісності даних, так і походження даних (як визначено в ISO 7498-2).

8.1.2 *Розповсюдження відкритих ключів, механізм 2*

Цей механізм передає до *B* інформацію щодо відкритого ключа суб'єкта *A* незахищеним каналом. Щоб перевірити цілісність і походження одержаної інформації щодо відкритого ключа використовують другий автентифікований канал. Такий механізм є корисним, коли інформацію щодо відкритого ключа РКІ пересилають широкосмуговим каналом, у той час як автентифікація інформації щодо відкритого ключа проводиться автентифікованим вузькосмуговим каналом таким, як телефон, кур'єр, реєстровна пошта. Додатковою вимогою є розподілення суб'єктами спільної ґешфункції *hash*, як визначено в ISO/IEC 10118-1. Повинні задовольнятися такі вимоги:

1. Інформація щодо відкритого ключа *A* — $РКІ_A$ — містить щонайменше розрізнювальний ідентифікатор суб'єкта *A* і його відкритий ключ. Додатково вона може містити номер серії, період дійсності, позначку часу та інші елементи даних.

2. У зв'язку з тим, що інформація щодо відкритого ключа РКІ не містить жодних таємних даних, канал взаємодії не повинен забезпечувати конфіденційність.

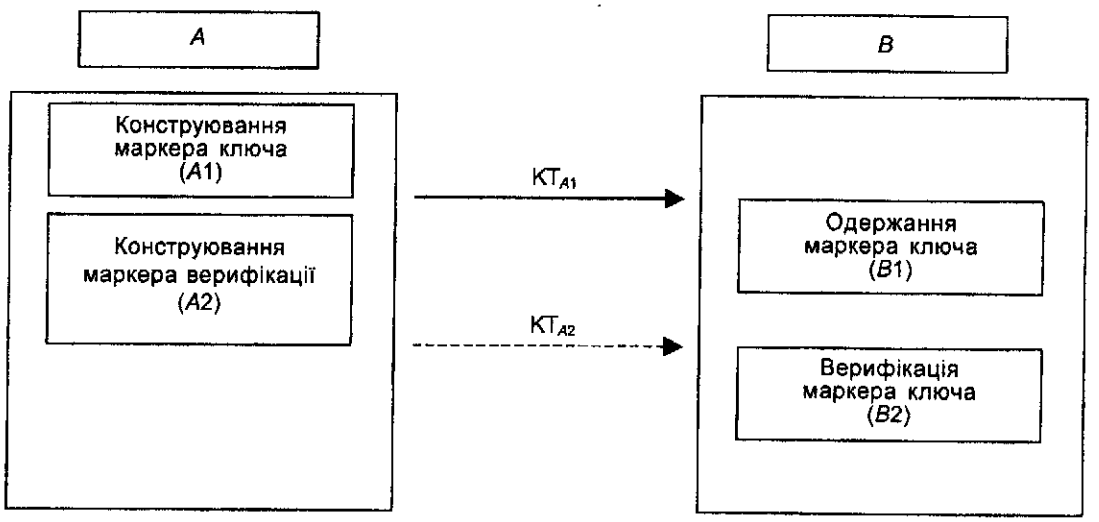

Рисунок 15 — Передавання відкритого ключа, механізм 2

Конструювання маркера ключа (*A1*)

A конструює маркер ключа KT_{A1}, який містить інформацію щодо відкритого ключа суб'єкта *A* і надсилає його до *B*.

$$KT_{A1} = PKI_A || Text1$$

Одержання маркера ключа (*B1*)

B одержує маркер ключа, відновлює PKI_A — інформацію щодо відкритого ключа суб'єкта *A*, необов'язково, верифікує ключ верифікації *A* і запам'ятовує його захищеним від втручання чином для подальшої верифікації і використання.

Конструювання маркера верифікації (*A2*)

A обчислює контролювальне значення *hash* (PKI_A) на його інформації щодо відкритого ключа і надсилає до *B*, із використанням другого незалежного і автентифікованого каналу (наприклад, кур'єра або реєстровної пошти), це контролювальне значення разом із необов'язковими розрізнювальними ідентифікаторами суб'єктів *A* і *B*.

$$KT_{A2} = A||B||hash(PKI_A)||Text2$$

Верифікація маркера ключа (*B2*)

Після одержання маркера верифікації KT_{A2} *B*, необов'язково, контролює розрізнювальні ідентифікатори *A* і *B*, обчислює контролювальне значення на інформації щодо відкритого ключа суб'єкта *A*, одержаній в маркері верифікації KT_{A1} і порівнює його з контролювальним значенням, одержаним у маркері верифікації KT_{A2}. Якщо верифікації успішні, *B* вносить відкритий ключ у перелік активних відкритих ключів (цей перелік повинен бути захищений від втручання).

Примітка. Цей механізм передавання відкритого ключа має такі властивості:

1. Цей механізм можна використовувати для пересилання відкритого ключа верифікації (для асиметричних систем підпису) або відкритого ключа зашифровування (для асиметричних систем шифрування), або відкритого ключа узгодження ключів.

2. Автентифікація в цьому контексті включає автентифікацію як цілісності даних, так і походження даних.

3. Якщо пересланий відкритий ключ є ключем для асиметричної системи підпису без відновлення повідомлень, *A* може підписати маркер KT_{A1} із використанням відповідного особистого ключа підпису. У цьому випадку верифікація підпису *A* на кроці (*B1*) із використанням одержаного відкритого ключа верифікації підтверджує, що *A* знає відповідний особистий ключ підпису і тому, можливо, був єдиним суб'єктом, який знає відповідний особистий ключ підпису на момент генерування маркера. Якщо в PKI використано позначку часу, то верифікація підтверджує, що *A* на поточний момент знає відповідний-особистий ключ підпису.

4. Для маркера верифікації можуть бути використані листи, підписані вручну.

8.2 Розповсюдження відкритих ключів із використанням третьої довірчої сторони

Автентифікація відкритого ключа суб'єкта може бути гарантована обміном відкритими ключами у формі сертифікатів відкритих ключів. Сертифікат відкритого ключа містить інформацію щодо відкритого ключа, разом із цифровим підписом, забезпеченим третьою довірчою стороною, повноважним сертифікатором (CA). Залучення CA зводить проблему автентифікованого розподілення

відкритого ключа користувача до проблеми автентифікованого розподілення відкритого ключа суб'єкта CA за рахунок довірчого центру (CA) (див. ISO/IEC 9594-8, 11770-1).

8.2.1 ***Розподілення відкритих ключів, механізм 3***

Цей механізм пересилає відкритий ключ від суб'єкта *A* до суб'єкта *B* автентифікованим чином. Він оснований на припущенні, що достовірний сертифікат відкритого ключа *CertA* PKI_A — інформації щодо відкритого ключа суб'єкта *A* — був виданий деяким повноважним сертифікатором і що *B* має доступ до автентифікованої копії відкритого перетворення верифікації V_{CA} цього повноважного сертифікатора CA, який видав сертифікат відкритого ключа.

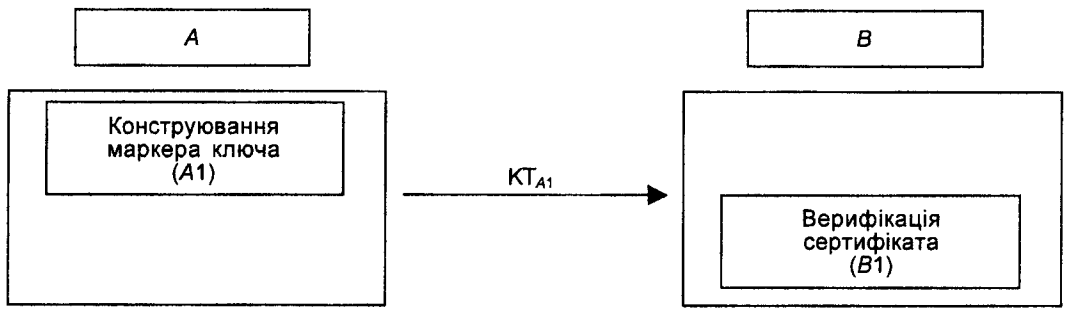

Рисунок 16 — Передавання відкритого ключа, механізм 3

Конструювання маркера ключа (*A*1)

A конструює маркер ключа KT_{A1}, який містить сертифікат відкритого ключа суб'єкта *A* і надсилає його до *B*.

Верифікація сертифіката (*B*1)

Після одержання сертифіката відкритого ключа, *B* використовує відкрите перетворення верифікації повноважного сертифікатора V_{CA} для верифікації автентичності інформації щодо відкритого ключа і контролю достовірності відкритого ключа суб'єкта *A*.

Якщо *B* хоче впевнитися в тім, що сертифікат відкритого ключа суб'єкта *A* ще не був відкликаний, *B* повинен проконсультуватися у третьої довірчої сторони (такої, як CA) по деякому автентифікованому каналу.

Примітка. Цей механізм передавання відкритого ключа має такі властивості:

1. Число проходів: 1. Але може вимагатися запит від *B* до *A* на передавання сертифіката відкритого ключа. Цей додатковий прохід є необов'язковим і тут не показаний. Сертифікат відкритого ключа суб'єкта *A* може також бути розподілений через довідник. У цьому випадку цей механізм передавання відкритого ключа повинен виконуватися між довідником і суб'єктом *B*.

2. Автентифікація суб'єкта: автентифікація суб'єкта цим механізмом не забезпечується.

3. Підтвердження ключа: одержання сертифіката відкритого ключа забезпечує підтвердження того, що відкритий ключ був сертифікований суб'єктом CA.

4. Відкритий ключ верифікації v_{CA} суб'єкта CA повинен ставати доступним для *B* автентифікованим чином. Це можна зробити, використавши механізми, наведені в розділі 8.

ДОДАТОК А
(довідковий)

ВЛАСТИВОСТІ МЕХАНІЗМУ ВСТАНОВЛЕННЯ КЛЮЧА

Наступні таблиці сумують головні властивості механізмів встановлення (передавання) ключа, визначені цією частиною стандарту.

Використані такі познаки:

A — механізм забезпечує властивість відносно суб'єкта *A;*

A, B — механізм забезпечує властивість відносно обох суб'єктів *A* і *B*;

Немає — механізм не забезпечує властивість;

Не обов. — механізм може забезпечити властивість як необов'язкову з використанням інших засобів;

(*A*) — механізм може, необов'язково, забезпечити властивість відносно суб'єкта *A* з використанням інших засобів.

Дії з відкритим ключем: кількість обчислень асиметричного перетворення, тобто «2,1» означає, що суб'єкт *A* потребує два обчислення функції *F*, а *B* потребує одне обчислення функції *F* у механізмі узгодження ключа 2.

Властивості механізму узгодження ключа

Механізм	1	2	3	4	5	6	7
Число проходів	0	1	1	2	2	2	3
Неявна автентифікація ключа	*A, B*	*B*	*A, B*	немає	*A, B*	*A, B*	*A, B*
Підтвердження ключа	немає	немає	*B*	немає	не обов.	не обов.	*A, B*
Керування ключем	*A, B*	*A, B*	*A, B*	*A, B*	*A, B*	*A, B*	*A, B*
Автентифікація суб'єкта	немає	немає	(*A*)	немає	немає	*B*	*A, B*
Операції з відкритим ключем	1,1	2,1	3 (або2),2	2,2	2,2	2,2	3,3

Властивості механізму передавання ключа

Механізм	1	2	3	4	5	6
Число проходів	1	1	1	2	3	3
Неявна автентифікація ключа	*B*	*B*	*B*	*A*	*A, B*	*A, B*
Підтвердження ключа	немає	*B*	*B*	*A*	(*A*), *B*	*A, B*
Керування ключем	*A*	*A*	*A*	*B*	*A* по відношенню *B*	(*A*), *B*
Автентифікація суб'єкта	немає	(*A*)	(*A*)	*B*	*A, B*	*A, B*
Дії з відкритим ключем	1.1	2.2	2.2	4.4	4.4	2.2

ДОДАТОК В
(довідковий)

ПРИКЛАДИ МЕХАНІЗМУ УЗГОДЖЕННЯ КЛЮЧА

Цей інформаційний додаток надає приклади деяких механізмів встановлення ключа, наведених у цій частині стандарту.

Спочатку визначають поширений приклад функції *F* і супутні множини *G* і *H*, відносно яких припускають, що вони задовольняють вимоги, перелічені у розділі 6, який обумовлює, що певні параметри обирають певним чином.

Нехай *p* є простим числом, *G* є множиною елементів поля Галуа з *p* елементами F_p і $H = \{1,\ldots, p–2\}$. Нехай *g* є основним елементом F_p, тоді множина

$$F(h, g) = g^h \bmod p.$$

F є комутативною по відношенню до *h*.

$$(g^{hB})^{hA} = (g^{hA})^{hB} = (g^{hA})^{hB} \bmod p.$$

Просте число *p* повинно бути досить великим, щоб *F*(*, *g*) могла розглядатися як односпрямована функція. Нехай кожен суб'єкт *X* має особистий ключ h_X в *H*, відомий тільки *X*, і відкритий ключ $p_X = g^{hX} \bmod p$, відомий всім іншим суб'єктам.

Примітка. Обрання параметрів.

Для дискретного логарифма за простим модулем: розмір простого числа треба обирати так, щоб обчислити дискретний логарифм у відповідній циклічній групі було неможливо. Деякі інші вимоги до простого числа можуть бути накладені з метою зробити дискретний логарифм не придатним.

Рекомендовано або обирати *p* суворо простим так, щоб *p* – 1 мав великий простий множник, або обирати *g*, який був би генератором групи великого простого порядку *q*.

Складення модуля для дискретного логарифма: модуль повинен бути обраний як добуток двох окремих непарних простих, які повинні триматися в таємниці. Розміри модулів треба обирати так, щоб розкладення їх на множники обчислюванням було неможливе. Можуть бути накладені деякі додаткові умови обрання простих чисел із метою зробити розкладення їх обчислюванням на множники неможливим.

B.1 Неінтерактивне узгодження ключів Діффі-Гелмана.

Це приклад [6] узгодження ключа, механізм 1.

Конструювання ключа (*A*1)

З використанням свого власного приватного ключа узгодження ключа h_A і відкритого ключа узгодження ключа p_B суб'єкта *B*, *A* обчислює розподілений ключ як

$$K_{AB} = p_B^{hA} \bmod p.$$

Конструювання ключа (*B*1)

З використанням свого власного приватного ключа узгодження ключа h_B і відкритого ключа узгодження ключа p_A суб'єкта *A*, *B* обчислює розподілений ключ як

$$K_{AB} = p_A^{hB} \bmod p.$$

B.2 Механізм, оснований на тотожності

Це приклад [8] узгодження ключа, механізм 1, який є основаним на тотожності в такому сенсі:

— відкритий ключ суб'єкта може бути відтворений з деякої комбінації його тотожності і його сертифіката;

— автентичність сертифіката не верифікують безпосередньо, але правильний відкритий ключ може бути відновленим лише з автентичного сертифіката.

Нехай (*n*, *y*) є відкритим ключем верифікації повноважного сертифікатора схеми цифрового підпису з відновленням повідомлення, визначеній в ISO/IEC 9796, додаток A (довідковий). Внаслідок цього, *n* є добутком двох великих простих чисел *p* і *q*, які тримають у таємниці, і *y* є простим з *l* cm ($p - 1$, $q - 1$).

Нехай *O* є цілим великого порядку модуля *n* і $g = O^y \bmod n$.

Нехай I_X є результатом додання надлишковості (згідно з ISO/IEC 9796) до відкритої інформації суб'єкта *X*, яка містить щонайменше розрізнювальний ідентифікатор *X* і, можливо, порядковий номер, період дійсності, позначку часу та інші елементи даних. Тоді пара керування ключем суб'єкта $X - (h_X, p_X)$, де h_X є цілим, меншим за *n*, і

$$p_x = g^{hx} (\bmod n).$$

Сертифікат обчислює повноважний сертифікатор як

$$Cert_x = s_x O^{hx} (\bmod n),$$

де s_X є таким цілим, що:

$$s_x^y I_x = 1 (\bmod n).$$

Конструювання ключа (*A*1)

A обчислює відкритий ключ *B* як

$$p_B = Cert_B^y \cdot I_B \bmod n$$

і обчислює розподілений таємний ключ як

$$K_{AB} = p_B^{hA} = g^{hA\ hB} \bmod n.$$

Конструювання ключа (*B*1)

B обчислює відкритий ключ *A* як

$$p_A = Cert_A^y \cdot I_A \bmod n$$

і обчислює розподілений таємний ключ як

$$K_{AB} = p_A^{hB} = g^{hA\ hB} \bmod n.$$

Примітка. Однопрохідний і двопрохідний механізми, основані на тотожності, з використанням однакових установок наведено в посиланнях [8], [19] і [20] додатка D.

B.3 Узгодження ключа за ЕльГамалем

Це приклад [7] узгодження ключа, механізм 2.

Зазначимо, що p повинне бути суворо простим числом і експоненти не мають форму 0, +1, –1 mod p.

Конструювання маркера ключа (*A*1)

A випадково і таємно генерує r у {1,..., p – 2}, обчислює $g^r \bmod p$ і конструює маркер ключа

$$KT_{A1} = g^r \bmod p$$

і надсилає його до *B*.

Конструювання ключа (*A*2)

A обчислює розподілений ключ

$$K_{AB} = (p_B)^r \bmod p = g^{hBr} \bmod p.$$

Конструювання ключа (*B*1)

B обчислює розподілений ключ

$$K_{AB} = (g^r)^{hB} = g^{hBr} \bmod p.$$

B.4 Узгодження ключа за Нібергом-Рюппелем

Це приклад [18] узгодження ключа, механізм 3. Систему підпису і систему узгодження ключа обирають так, що систему підпису визначають парою (h_X, p_X).

Нехай q є великим простим дільником числа p – 1, g елемент F_P порядка q і множина H = {1,..., q – 1}. Тоді несиметрична пара ключів суб'єкта X, яку використовують для підпису, є (h_X, p_X), де h_X є елементом H і

$$p_x = g^{hx} \bmod p.$$

Щоб запобігти повторенню старого маркера ключа, цей механізм використовує позначку часу або порядковий номер, *TVP*, і криптографічну ґеш-функцію *hash*, яка відображає рядки бітів довільної довжини у випадкові числа великої підмножини {1,..., p – 1}, наприклад, *H*.

Конструювання ключа (*A*1.1)

A випадково і таємно генерує r в *H* і обчислює

$$e = g^r \bmod p.$$

Додатково *A* обчислює розподілений таємний ключ, як

$$K_{AB} = p_B^r \bmod p.$$

З використанням розподіленого таємного ключа K_{AB}, *A* обчислює криптографічне контролювальне значення розрізнювального ідентифікатора посилача *A* і порядковий номер або часову позначку *TVP*.

$$e' = e\,hash(K_{AB}||A||TVP) \bmod p.$$

Підпис маркера ключа (*A*1.2)

A обчислює підпис

$$y = r - h_A e' \bmod q.$$

A формує маркер ключа

$$KT_{A1} = A||e||TVP||y$$

і надсилає його до *B*.

Конструювання ключа (*B*1.1)

B обчислює розподілений таємний ключ із використанням свого особистого ключа узгодження ключа h_B

$$K_{AB} = e^{hB} \bmod p.$$

З використанням розподіленого таємного ключа K_{AB} *B* обчислює криптографічне контролювальне значення розрізнювального ідентифікатора посилача *A* і *TVP*, і обчислює

$$e' = e\,hash(K_{AB}||A||TVP) \bmod p.$$

Верифікація підпису (*B*1.2)

B перевіряє дійсність *TVP* і з використанням відкритого ключа p_A посилача верифікує рівність

$$e = g^{y} p_A^{e'} \bmod p.$$

B.5 Узгодження Діффі-Гелмана

Це приклад [6] узгодження ключа, механізм 4.

Зазначимо, що p повинне бути суворо простим числом і експоненти не мають форму 0, +1, –1 mod$p$.

Конструювання маркера ключа (*A*1)

A випадково і таємно генерує r_A в {1,..., p – 2}, обчислює $g^{rA} \bmod p$, конструює маркер ключа

$$\mathrm{KT}_{A1} = g^{rA} \bmod p$$

і надсилає його до *B*.

Конструювання маркера ключа (*B*1)

B випадково і таємно генерує r_B в {1,..., p – 2}, обчислює $g^{rB} \bmod p$, конструює маркер ключа

$$\mathrm{KT}_{B1} = g^{rB} \bmod p$$

і надсилає його до *A*.

Конструювання ключа (*A*2)

A обчислює розподілений ключ

$$K_{AB} = (g^{rB})^{rA} = g^{rA\,rB} \bmod p.$$

Конструювання ключа (*B*2)

B обчислює розподілений ключ

$$K_{AB} = (g^{rA})^{rB} = g^{rA\,rB} \bmod p.$$

B.6 Узгодження ключа A(0) за Макумото-Такашіма-Імаі

Це приклад [1] узгодження ключа, механізм 5.

Одним із рекомендованих методів є використання суворо простого числа і контроль, що експоненти не мають форму 0, +1, –1 mod$p$.

Конструювання маркера ключа (*A*1)

A випадково і таємно генерує r_A в {1,..., p – 2}, обчислює маркер ключа

$$\mathrm{KT}_{A1} = g^{rA} \bmod p$$

і надсилає його до *B*.

Конструювання маркера ключа (*B*1) *B* випадково і таємно генерує r_B в {1,..., p – 2}, обчислює маркер ключа

$$\mathrm{KT}_{B1} = g^{rB} \bmod p$$

і надсилає його до *A*.

Конструювання ключа (*B*2)

B обчислює розподілений ключ як

$$K_{AB} = \omega(\mathrm{KT}_{A1}^{hB}, p_A^{rB}) = \mathrm{KT}_{A1}^{hB}, p_A^{rB} \bmod p.$$

Конструювання ключа (*A*2)

A обчислює розподілений ключ

$$K_{AB} = \omega(p_B^{rA}, \mathrm{KT}_{B1}^{hA}) = \mathrm{KT}_{A1}^{hB}, p_A^{rB} \bmod p.$$

B.7 Протокол Беллера-Якобі

Ця частина додатка надає опис оригінального протоколу Беллера-Якобі [4], який треба використовувати для виведення механізму узгодження ключа 6.

Примітка. Механізм не повністю сумісний з механізмом 6 після його оптимізації для певних ситуацій. Він використовує схему підпису ЕльГамаля, а також використовує додатковий алгоритм симетричного шифрування для конфіденційного пересилання суб'єкту *A* ключа верифікації підпису суб'єкта *B* і його сертифіката, чим забезпечує анонімність.

Нехай *enc*: $KXM \rightarrow C$ є узгодженою криптографічною функцією, такою як DES, де K = поле ключа, M = поле повідомлень і C = поле криптограм.

Нехай S_X означує дію суб'єкта X підписування за ЕльГамалем. Процес, наведений нижче, підкреслює різницю між автономними і не автономними операціями, необхідними родині схем підпису за ЕльГамалем.

P_X і C_X використовують для зазначення, відповідно, відкритого ключа і сертифіката суб'єкта X. Відкрите перетворення зашифровування суб'єкта X (яке використовує P_X), зазначене E_X (модульне піднесення до квадрату у випадку Рабіна).

Автономне обчислювання: B підбирає випадкове число r_B і обчислює

$$u = g^{rB} \bmod p.$$

Конструювання маркера ключа (*A*1)

A підбирає випадкове число r_A і обчислює

$$KT_{A1} = (r_A||A||C_A)$$

і надсилає його до B.

Оброблення маркера ключа (*B*1)

B виробляє підпис

$$BS = (u, v) = S_B(r_A||A).$$

Потім B підбирає випадкове число і формує

$$KT_{B1} = E_A(BS)||enc(u,(B||P_B||C_B||x_B))$$

і надсилає його до A.

Конструювання ключа (*B*2)

Розподілений таємний ключ складається з частини підпису B, u.

Автентифікація суб'єкта і конструювання ключа (*A*2)

A розшифровує маркер ключа $E_A(BS)$, щоб знайти сесійний ключ u, потім розшифровує узгоджене зашифровування

$$enc(u,(B||P_B||C_B||x_B))$$

з використанням сесійного ключа u, щоб знайти ідентифікатор, відкритий ключ, і сертифікат передбачуваної сторони B. A верифікує сертифікат C_B і, якщо результат позитивний, використовує функцію верифікації V_B для верифікації підпису BS суб'єкта B. Якщо верифікація успішна, A приймає u як розподілений таємний ключ.

B.8 Пересилання ключа за ЕльГамалем

Це приклад [7] передавання ключа механізму 1.

Обирають і роблять відкритими необхідне просте число p і генератор g із Z_P^*. Приватний і відкритий ключі узгодження ключа суб'єкта B, відповідно, h_B і

$$p_B = g^{hB} \bmod p.$$

Конструювання маркера ключа (*A*1)

A отримав ключ K (в межах $0 < R < p$) і хоче переслати його надійно до B. A випадково і таємно генерує випадкове ціле r, $(1 < r < p)$ і зашифровує K як

$$BE = K\,(p_B)^r \bmod p.$$

Потім A конструює маркер ключа

$$KT_{A1} = BE||g^r \bmod p$$

і надсилає його до B.

Розібрання маркера ключа (*B*1)

B відновлює ключ K із використанням свого приватного ключа узгодження ключа h_B, обчислюючи

$$K = BE \cdot (g^r)^{-hB} \bmod p.$$

B.9 Пересилання ключа з підписом автора за ЕльГамалем

Це приклад передавання ключа механізму 2. Обирають і роблять відкритими підхоже просте

число p і генератор g із Z_P^*. Приватний і відкритий ключі узгодження ключа суб'єкта B, відповідно, h_B і

$$p_B = g^{hB} \bmod p.$$

Приватне і відкрите перетворення підписування суб'єкта A означують, відповідно, S_A і V_A; (S_A і V_A) може означувати довільну систему підпису, наприклад, підпис RSA і верифікацію підпису, яка визначена в ISO/IEC 9796.

Шифрування ключа (*A*1.1)

A отримав ключ K і хоче його надійно переслати до B. A випадково і таємно генерує випадкове число r в $\{1,..., p-2\}$ і зашифровує блок даних ключа $A||K$ як

$$\mathrm{BE} = (A||K)\cdot(p_B)^r \bmod p.$$

Зазначимо, що K треба обирати так, щоб значення $(A||K)$ було меншим за просте число p.

Конструювання маркера ключа (*A*1.2)

A формує блок даних маркера, який складається з розрізнювального ідентифікатора одержувача B, і необов'язкової позначки часу або порядкового номера TVP, g^r і зашифрованого блока BE. Потім A підписує блок даних маркера з використанням свого приватного перетворення підписування S_A і надсилає остаточний маркер ключа

$$\mathrm{KT}_{A1} = S_A(B||TVP||g^r||\mathrm{BE})$$

до B.

Верифікація маркера ключа (*B*1.1)

B використовує відкрите перетворення верифікації посилача V_A для верифікації цифрового підпису одержаного маркера ключа KT_{A1}. Потім B контролює ідентифікатор одержувача B і, необов'язково, TVP.

Розшифровування ключа (*B*1.2)

B розшифровує блок BE із використанням приватного ключа узгодження ключа h_B обчислюванням

$$A||K = \mathrm{BE}\cdot(g^r)^{-hB} \bmod p.$$

Потім B контролює ідентифікатор посилача A. Якщо всі перевірки успішні, B приймає ключ K.

B.10 Пересилання ключа за RSA

Це приклад передавання ключа механізму 1. Несиметрична система шифрування (E_B, D_B) суб'єкта B складається з модуля RSA $n = pq$, із відкритою експонентою e і приватною експонентою d, таких, що $ed = 1 \bmod (p-1)(q-1)$. Припускається, що A має автентичну копію параметрів шифрування (e, n) суб'єкта B.

Конструювання маркера ключа (*A*1)

A отримує ключ K для пересилання до B. Припускається, що *Text1*, *Text2* і, необов'язковий, TVP всі дорівнюють нулю (тобто опущені). Додатково припускають, що дані відформатовані адекватно до виконання алгоритмів RSA (тобто містять певну надлишковість). A створює і надсилає до B блок даних

$$\mathrm{KT}_{A1} = E_B(A||K) = (A||K)^e \bmod n.$$

Розібрання маркера ключа (*B*1)

B одержує ці дані і обчислює

$$(\mathrm{KT}_{A1})^d \bmod n = (A||K).$$

Одержувач B може відрізнити це повідомлення від випадкового повідомлення, контролюючи певні умови надлишковості у вмісті повідомлення $A||K$.

Припустимо також, що тотожність A в цьому відновленому повідомленні має деяку надлишковість, яку можна перевірити, або очікуваний формат, B контролює, що відновлений ідентифікатор A має очікувану форму, і приймає повідомлення лише в разі успішності перевірки.

ДОДАТОК С
(довідковий)

ПРИКЛАДИ ВСТАНОВЛЕННЯ КЛЮЧА, ЯКІ ҐРУНТУЮТЬСЯ НА ЕЛІПТИЧНИХ КРИВИХ

Метою цього додатка є показати, як механізм встановлення ключа, наведений в цій частині стандарту, може бути реалізований в термінах еліптичних кривих. Обрання представлених протоколів широко використовують в додатку В.

Математичне підґрунтя для еліптичних кривих:

Еліптична крива E є несингулярна кубічна крива, визначена на деякому полі K. Еліптична крива може бути наведена як множина рішень (x, y) $(x, y \in K)$ рівняння

$$Y^2 = X^3 + aX + b$$

разом з особливою точкою q — точкою в нескінченності.

Еліптичні криві наділені властивостями бінарної дії $\circ$: $E \cdot E \rightarrow E$, приєднуючи до кожної пари $(P1, P2)$ точок в E третю точку $P1 \circ P2$. Відносно цієї дії E є Абелевою групою з нейтральним елементом q.

Нехай P є деякою точкою на еліптичній кривій E, яка генерує циклічну групу $<P>$ із граничною чисельністю q відносно групової дії "$\circ$". Тоді кожен елемент із $<P>$ є деяким ступенем $P^{[k]}$ від P, де $P^{[k]}$ є абревіатурою k-кратної операції $(P \circ P \circ \ldots \circ P)$.

Дискретне піднесення до степеня $F(\ , P)$ на $<P>$ визначають як

$$F(k, P) = P^{[k]}, \text{ для } k \in \{1, \ldots, q - 1\}.$$

Зазначимо, що для довільних $h, k \in \{1, \ldots, q - 1\}$ виконується рівняння

$$(P^{[h]})^{[k]} = P^{[h]\cdot[k]} = (P^{[k]})^{[h]},$$

якщо група $<P>$, генерована P, є Абелевою.

З іншого боку, якщо задана довільна точка $Q \in <P>$, то однозначно визначене ціле $x \in \{1, \ldots, q - 1\}$ таке, що $Q = P^{[x]}$ означується як дискретний логарифм Q з основою P.

Криптографічна важливість еліптичних кривих спирається на передбачувані труднощі визначення дискретного логарифма на еліптичній кривій, визначеній на скінченних полях, які — за наявних знань — значно більші за розкладення на множники цілого числа або обчислювання дискретного логарифма в $GF(p)$. Це надає можливість використовувати системи з відкритими ключами, основані на еліптичних кривих, із значно меншим числом параметрів, порівняно з більш знайомими системами з відкритим ключем.

Система познак:

У цьому додатку будемо дотримуватися термінології, введеної в попередньому розділі.

Крім того, зафіксуємо такі познаки:

K є скінченним полем, яке складається з точно p^n елементів, де p є простим числом, більшим за 3, а n — додатнє ціле.

E є еліптична крива на K і P є точкою на E, яка генерує циклічну групу $<P>$ із чисельністю q. Припускаємо, що q є простим числом і множина $H = \{1, \ldots, q - 1\}$.

Кожний суб'єкт X має особистий ключ h_X у H, відомий лише X, і відкритий ключ $P_X = G^{[hX]}$, відомий всім іншим суб'єктам.

Зазначимо, що особисті ключі є суворо простими цілими, в той час як таємні — це точки на кривій. Цим система відрізняється від систем із відкритим ключем, основаним на дискретних логарифмах із простим модулем, де обидва ключі є об'єктами того самого типу. Ця різниця між двома типами ключів у випадку еліптичних кривих є причиною того, чому необхідно ввести додаткову функцію, яка відображає точки $<P>$ в цілі в H, якщо хтось бажає перевести протоколи додатка В в еліптичні криві.

Нехай π: $<P> \rightarrow H$ є функцією такою, що об'єднання π і $F(\circ, P)$, задане як

$$k \rightarrow P^{[k]} \rightarrow \pi(P^{[k]})$$

односпрямоване.

Примітка 1. Критичним параметром для безпеки системи із відкритим ключем на основі еліптичних кривих є величина простого числа q. Ціле q повинно бути достатньо великим, щоб $F(\circ,P)$ могла вважатися односпрямованою функцією. Враховуючи алгоритми, які існують на тепер, $F(\circ,P)$ є односпрямованою, якщо $q > 2^{120}$.

Примітка 2. На відміну від систем, основаних на GF$(p)^*$ дискретних логарифмах (подібній до DSA), є можливість обрати параметри q і p^n приблизно однієї величини.

Примітка 3. Є ще декілька умов відносно простих p і q (наприклад, $p \neq q$) і параметрів кривої a і b, яких необхідно дотримуватися, щоб зробити обчислення дискретного логарифма на еліптичній кривій неможливим.

Примітка 4. Є багато можливостей визначити π. Одним простим методом є виконати проекцію точок із $<P>$ на їх x-координати і «читати» ці елементи поля, як цілі по mod q.

С.1 Неінтерактивне узгодження ключа типу Діффі-Гелмана

Це приклад узгодження ключа механізму 1.

Конструювання ключа (*A*1)

З використанням свого власного особистого ключа узгодження ключів h_A і відкритого ключа узгодження ключів P_B A обчислює розподілений ключ як

$$K_{AB} = (P_B)^{[hA]}.$$

Конструювання ключа (*B*1)

З використанням свого власного особистого ключа узгодження ключів h_B і відкритого ключа узгодження ключів P_A B обчислює розподілений ключ як

$$K_{AB} = (P_A)^{[hB]}.$$

С.2 Узгодження ключа типу ЕльГамаля

Це приклад узгодження ключа механізму 2.

Конструювання маркера ключа (*A*1)

A випадково і таємно генерує r в H, обчислює $(P_B)^{[r]}$, конструює маркер ключа

$$KT_{A1} = (P)^{[r]}$$

і надсилає його до B.

Конструювання ключа (*A*2)

A обчислює розподілений ключ

$$K_{AB} = (P_B)^{[r]} = P^{[hB \cdot r]}$$

Конструювання ключа (*B*1)

B обчислює зі своїм власним особистим ключем розподілений ключ із KT_{A1} так

$$K_{AB} = (KT_{A1})^{[hB]} = (P^{[r]})^{[hB]} = P^{[r \cdot hB]}.$$

С.3 Узгодження ключа за Нюберг-Рюппелем.

Це приклад узгодження ключа механізму 3.

Протокол не є копією один до одного протоколу ВЗ, але він слідує суттєвим ідеям ВЗ.

Систему підпису і систему узгодження ключів обирають так, щоб система підпису визначалася ключами (h_X, P_X).

Щоб запобігти повторенню старих маркерів ключів, цей приклад використовує позначку часу або порядковий номер TVP, і криптографічну ґеш-функцію hash, яка відображає рядки бітів довільної довжини у випадкові цілі, наприклад в H.

Конструювання ключа (*A*1.1)

A випадково і таємно генерує r в H і обчислює

$$R = P^{[r]}.$$

Додатково, A обчислює розподілений таємний ключ як

$$K_{AB} = (P_B)^{[r]}.$$

З використанням розподіленого таємного ключа K_{AB}, A обчислює криптографічне контролювальне значення у точці R, на розрізнювальному ідентифікаторі посилача A і порядковому номері або позначці часу TVP

$$e = hash(R||K_{AB}||A||TVP).$$

Підпис маркера ключа (*A*1.2)

A обчислює підпис

$$y = (r - h_A \cdot e) mod\, q$$

формує маркер ключа

$$KT_{A1} = (R||A||TVP||y)$$

і надсилає його до *B*.

Конструювання ключа (*B*1.1)

З використанням свого особистого ключа узгодження ключа h_B, *B* обчислює розподілений таємний ключ

$$K_{AB} = R^{[hB]}.$$

З використанням розподіленого таємного ключа K_{AB}, *B* обчислює криптографічне контролювальне значення на розрізнювальному ідентифікаторі *A* та *TVP* і обчислює

$$e = hash(R||\ K_{AB}||A||TVP).$$

Верифікація підпису (*B*1.2)

B контролює достовірність *TVP* і верифікує з використанням відкритого ключа посилача P_A тотожність

$$R = P^{[y]} \circ (P_A)^{[e]}$$

C.4 Узгодження ключа типу Діффі-Гелмана

Це приклад узгодження ключа механізму 4.

Конструювання маркера ключа (*A*1)

A випадково і таємно генерує r_A в *H* і обчислює $P^{[rA]}$, конструює маркер ключа

$$KT_{A1} = P^{[rA]}$$

і надсилає його до *B*.

Конструювання маркера ключа (*B*1)

B випадково і таємно генерує r_B в *H*, обчислює $P^{[rB]}$, конструює маркер ключа

$$KT_{B1} = P^{[rB]}$$

і надсилає його до *A*.

Конструювання ключа (*A*2)

A обчислює розподілений ключ як

$$K_{AB} = (P^{[rB]})^{[rA]} = P^{[rB]\cdot[rA]}.$$

Конструювання ключа (*B*2)

B обчислює розподілений ключ як

$$K_{AB} = (P^{[rA]})^{[rB]} = P^{[rA]\cdot[rB]}.$$

C.5 Узгодження ключа типа A(0) Мацумото-Такашима-Імаі

Це приклад узгодження ключа механізму 5.

Конструювання маркера ключа (*A*1)

A випадково і таємно генерує r_A в *H*, обчислює маркер ключа

$$KT_{A1} = P^{[rA]}$$

і надсилає його до *B*.

Конструювання маркера ключа (*B*1)

B випадково і таємно генерує r_B в *H*, обчислює маркер ключа

$$KT_{B1} = P^{[rB]}$$

і надсилає його до *A*.

Конструювання ключа (*B*2)

B обчислює розподілений ключ як

$$K_{AB} = \omega(KT_{A1}^{[hB]}, P_A^{[rB]}),$$

де ω односпрямована функція.

Конструювання ключа (*A*2)

A обчислює розподілений ключ як

$$K_{AB} = \omega(\mathrm{KT}_{B1}^{[hA]}, P_B^{[rB]}).$$

С.6 Пересилання ключа типу ЕльГамаля

Це приклад передавання ключа механізму 1.

Конструювання маркера ключа (*A*1)

A отримав ключ $K \in H$ і хоче передати його надійно до *B*.

A випадково і таємно генерує ціле $r \in H$, обчислює точку на кривій $P^{[r]}$ і зашифровує *K* як

$$\mathrm{BE} = (K \cdot \pi((P_B^{[r]})) \bmod q.$$

Потім *A* конструює маркер ключа

$$\mathrm{KT}_{A1} = \mathrm{BE} || P^{[r]}$$

і надсилає його до *B*.

Розібрання маркера ключа (*B*1)

Щоб відновити ключ *K*, суб'єкт *B* визначає з $(P^{[r]})$ із використанням свого особистого ключа узгодження ключа h_B, точку на кривій $(P_B)^{[r]} = (P^{[r]})^{[hB]}$ і на наступному — проекцію $\pi((P_B)^{[r]})$.

Остаточно *B* отримує ключ *K* обчисленням

$$K = (\mathrm{BE}) \cdot (\pi((P_B^{[r]})^{[hB]}))^{-1} \bmod q.$$

С.7 Пересилання ключа типу ЕльГамаля з підписом автора

Це приклад передавання ключа механізму 2. Особистий і відкритий ключі узгодження ключів суб'єкта *B* є, відповідно, h_B і

$$P_B = (P)^{[hB]}.$$

Особисте і відкрите перетворення підписування суб'єкта *A*, відповідно, означимо S_A і V_A; (S_A, V_A) можуть означувати будь-яку систему підпису, наприклад систему підпису, визначену в ISO/IEC 9796.

Зашифровування ключа (*A*1.1)

A отримав ключ *K* і хоче переслати його надійно до *B*. *A* випадково і таємно генерує ціле $r \in H$, точки на кривій $P^{[r]}$, $P_B^{[r]}$ і зашифровує блок даних ключа $A||K$ як

$$\mathrm{BE} = (A||K) \cdot (\pi((P_B^{[r]})) \bmod q.$$

Зазначимо, що *K* треба обирати так, щоб значення $(A||K)$ було меншим за просте число *q*.

Конструювання маркера ключа (*A*1.2)

A формує блок даних маркера, який складається з розрізнювального ідентифікатора одержувача *B*, необов'язкової позначки часу або порядкового номера *TVP* і зашифрованого блоку BE. Потім *A* підписує блок даних маркера з використанням свого приватного перетворення підписування S_A і надсилає остаточний маркер ключа

$$\mathrm{KT}_{A1} = (B||TVP||P^{[r]}||\mathrm{BE})$$

і його підпис

$$S_A(B||TVP||P^{[r]}||\mathrm{BE})$$

до *B*.

Верифікація маркера ключа (*B*1.1)

B використовує відкрите перетворення верифікації посилача V_A для верифікації цифрового підпису одержаного маркера ключа KT_{A1}. Потім *B* контролює ідентифікацію одержувача *B* і, необов'язково, *TVP*.

Розшифровування ключа (*B*1.2)

B розшифровує блок BE з використанням свого особистого ключа узгодження ключа h_B обчисленням

$$(A||K) = (\mathrm{BE}) \cdot (\pi((P_B^{[r]})^{[hB]}))^{-1} \bmod q.$$

Потім *B* контролює ідентифікацію посилача *A*. Якщо всі перевірки успішні, *B* приймає ключ *K*.

ДОДАТОК D
(довідковий)

БІБЛІОГРАФІЯ

1 ANSI X9.30 199x, Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry, Part 1: The Digital Signature Algorithm (DSA)

2 ANSI X9.30 199x, Public Key Cryptography Using Irreversible Algorithms for the Financial Services Industry, Part 3: Certificate Management for DSA

3 ANSI X9.31 199x, Public key cryptography using reversible algorithms for the financial services industry — Pan 4: Management of symmetric algorithm keys using RSA

4 Beller M.J., Yacobi Y., Fully-fledged two-way public authentication and key agreement for low-cost terminals. Electronic Letters, Vol. 29, no. 11 (27 May 93), pp 999—1001

5 RIPE, Integrity Primitives for Secure Information Systems — Final Report of RACE Integrity Primitives Evaluation (RIPE-RACE 1040), LNCS 1007, A. Bos-selaers, B. Preneel, Eds., Springer-Verlag, 1995

6 Diffie W., Hellman M.E., New Directions in Cryptography, IEEE Trans. on Inform. Theory, vol. IT-22, pp. 644—654, Nov. 1976

7 ElGamal, T., A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, IEEE Trans. on Inform. Theory, vol. IT-31, pp. 469—472, July 1985

8 Girault M., Pailles J.C., An identity-based scheme providing zero-knowledge authentication and authenticated key exchange. Proceedings of ESORICS 90, pp. 173—184

9 ISO 8732:1988, Banking — Key Management (Wholesale)

10 ISO/IEC 9594-8:1990, (CCITT X.509), Information technology — Open Systems Interconnection — The Directory — Authentication framework

11 ISO/IEC 9796:1991 Information technology — Security techniques — Digital signature scheme giving message recovery

12 ISO/IEC 9797:1994 Information technology — Security techniques – Data integrity mechanism using a cryptographic check function employing a block cipher algorithm

13 ISO/IEC 10118-2:1994 Information technology — Security techniques — Hash-functions — Part 2: Hash-functions using an n-bit block cipher algorithm

14 ISO/IEC 10118-3:1998 Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash functions

15 ISO/IEC 10118-4:1998 Information technology — Security techniques — Hash-functions — Part 4: Mechanisms using modular arithmetic

16 ISO 11166-1:1994 Banking — Key Management by means of asymmetric algorithms — Part 1: Principles, Procedures and Formats

17 Matsumoto T., TakashimaY., Imai H., On Seeking Smart Public-Key-Distribution Systems, Trans. of the IECE of Japan, vol.E69 no.2, Feb. 1986 pp. 99—106

18 Menezes A., Elliptic Curve Public Key Cryto-systems, Kluwer Academic Publishers, 1993

19 Nyberg K., Rueppel R.A., Message Recovery for Signature Schemes Based on the Discrete Logarithm Problem, Proceedings of Eurocrypt'94, Springer-Verlag, 1994

20 Okamoto E., Proposal for identity-based key distribution system. Electronic Letters, Vol. 22, n°24, 20 Nov. 86, pp. 1283—4

21 Tanaka K., Okamoto E., Key distribution system for mail systems using ID-related information directory, Computers & Security, Vol.10, 1991, pp. 25—33

22 ISO/IEC 10118-2:1994 Information technology — Security techniques — Hash-functions — Part 2: Hash-functions using an n-bit block cipher algorithm

23 ISO/IEC 10118-3:1998 Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash functions

24 ISO/IEC 10118-4:1998 Information technology — Security techniques — Hash-functions — Part 4: Mechanisms using modular arithmetic

25 ISO 11166-1:1994 Banking — Key Management by means of asymmetric algorithms — Part 1: Principles, Procedures and Formats

26 Matsumoto T., TakashimaY., Imai H., On Seeking Smart Public-Key-Distribution Systems, Trans. of the IECE of Japan, vol.E69 no.2, Feb. 1986 pp.99—106

27 Menezes A., Elliptic Curve Public Key Cryto-systems, Kluwer Academic Publishers, 1993

28 Nyberg K., Rueppel R.A., Message Recovery for Signature Schemes Based on the Discrete Logarithm Problem, Proceedings of Eurocrypt'94, Springer-Verlag, 1994

29 Okamoto E., Proposal for identity-based key distribution system. Electronic Letters, Vol. 22, n°24, 20 Nov. 86, pp. 1283—4

30 Tanaka K., Okamoto E., Key distribution system for mail systems using ID-related information directory, Computers & Security, Vol.10, 1991, pp. 25—33.

ДОДАТОК Е
(довідковий)

ПАТЕНТНА ІНФОРМАЦІЯ

За час розроблення цього стандарту [частина 3 ISO/IEC 11770], була отримана інформація відносно патентів, від яких може залежати його впровадження. Відповідні патенти визначені, як це наведено в таблиці, що наведена нижче. Очевидно ISO/IEC не може дати повноважну або вичерпну інформацію відносно правильності або сфери застосування цих патентів.

Визначені власники патентів підтвердили, що ліцензія на впровадження цього стандарту буде видана у відповідні терміни, забезпечено, що з тими, хто хоче придбати ліцензію, буде укладене взаємне узгодження.

Подальша інформація доступна зі сторони визначених власників патентів.

Галузь	Автор	№ патенту	Дата реєстрації	Контактна адреса
Diffe-Hellman key agreement	Hellman-Diffe-Merkle	US 4,200,770	1980-04-29	
RSA system	Rivest-Shamir-Adelman	US 4,405,829	1983-09-20	RSA data security, Inc. Director of Licensing 2955 Campus Drive, Suite 400 San Mateo, CA 94403-2507, USA
ID based DH key agreement	Eiji Okamono	JP 1871933 US 4876716 EP 0257585 CA 1279709	1994-09-26 1989-10-24 1992-11-25 1991-01-01	NEC Corporation Intellectual Property Division 7-1, Shiba 5-Chome, Mintao-Ku Tokyo 108-8001, Japan
Goss key agreement	Goss	US 4,956,863	1990-09-11	Jones Futurex™ Ink. Chief Operatinf Officer 3715 Atherton Road Rocklin, CA 95765, USA
ID based DH key agreement	Eiji Okamono-Kazue Tanaka	JP 1871933 US 4876716 EP 0257585 CA 1279709 AS 618229	1998–01-09 1991-07-02 1997-08-06 1995-01-03 1992-05-04	NEC Corporation Intellectual Property Division 7-1, Shiba 5-Chome, Mintao-Ku Tokyo 108-8001, Japan
Nyberg-Rueppel key agreement	Nyberg-Rueppel	US 5,600,725 EP 0,639,907	1997-02-04 відкритий	Certicom Corp. 200 Matheson Blvd. West Missisauga, Ontario, Canada L5R 3L7 R3 Security Engineering AG CH-8302 Glattzentrum, Switzerland

УКНД 35.040

Ключові слова: кодування інформацією, оброблення даних, інформаційний обмін, відсилання даних, захист інформації, методи безпеки, автентифікація, керування ключами.

Редактор **С. Мельниченко**
Технічний редактор **О. Касіч**
Коректор **Н. Тонишева**
Верстальник **В. Перекрест**

Підписано до друку 12.06.2007. Формат 60×84 1/8.
Ум.друк.арк. 4,65. Зам. 1970 Ціна договірна.

Відділ редагування нормативних документів ДП «УкрНДНЦ»
03115, м. Київ, вул. Святошинська, 2