TCVN **TIÊU CHUẨN QUỐC GIA**

TCVN 7817 – 4:2010

ISO/IEC 11770 – 4:2006

Xuất bản lần 1

CÔNG NGHỆ THÔNG TIN – KỸ THUẬT AN NINH – QUẢN LÝ KHÓA – PHẦN 4: CƠ CHẾ DỰA TRÊN BÍ MẬT YẾU

Information technology – Security techniques – Key management
Part 4: Mechanisms based on weak secrets

HÀ NỘI – 2010

Mục lục **Trang**

Lời nói đầu

TCVN 7817 - 4 : 2010 hoàn toàn tương đương với ISO/IEC 11770 - 4 : 2006.

TCVN 7817 - 4 : 2010 do Tiểu ban Kỹ thuật Tiêu chuẩn quốc gia TCVN/JTC1/SC27 "*Kỹ thuật mật mã*" biên soạn, Tổng cục Tiêu chuẩn Đo lường Chất lượng đề nghị, Bộ Khoa học và Công nghệ công bố.

Bộ tiêu chuẩn TCVN 7817 bao gồm các TCVN sau:

- TCVN 7817 – 1:2007 Công nghệ thông tin – Kỹ thuật mật mã – Quản lý khóa, phần 1: Khung tổng quát.
- TCVN 7817 – 2:2010 Công nghệ thông tin – Kỹ thuật an ninh – Quản lý khóa, phần 2: Cơ chế sử dụng kỹ thuật đối xứng.
- TCVN 7817 – 3:2007 Công nghệ thông tin – Kỹ thuật mật mã – Quản lý khóa, phần 3: Các cơ chế sử dụng kỹ thuật không đối xứng.
- TCVN 7817 – 4:2010 Công nghệ thông tin – Kỹ thuật an ninh – Quản lý khóa, phần 4: Cơ chế dựa trên bí mật yếu.

Công nghệ thông tin – Kỹ thuật an ninh – Quản lý khóa

Phần 4: Cơ chế dựa trên bí mật yếu

Information technology – Security techniques – Key management

Part 4: Mechanisms based on weak secrets

1 Phạm vi áp dụng

Tiêu chuẩn này quy định các cơ chế thiết lập khóa dựa trên bí mật yếu, tức là bí mật được con người ghi nhớ dễ dàng và do đó nó được chọn từ tập các khả năng khá nhỏ. Tiêu chuẩn này quy định các kỹ thuật mật mã thiết lập một hoặc nhiều khóa bí mật dựa trên bí mật yếu dẫn xuất từ một mật khẩu đã nhớ, trong khi ngăn chặn tấn công vét cạn ngoại tuyến liên quan đến bí mật yếu. Cụ thể là các cơ chế này được thiết kế để đạt được một trong ba mục đích sau:

1) **Thỏa thuận khóa được xác thực bằng mật khẩu cân bằng**: thiết lập một hoặc nhiều khóa bí mật giữa hai thực thể dùng chung bí mật yếu. Tại cơ chế thỏa thuận khóa được xác thực bằng mật khẩu cân bằng, các khóa bí mật dùng chung là kết quả của việc trao đổi dữ liệu giữa hai thực thể, tại đó khóa bí mật dùng chung được thiết lập với điều kiện hai thực thể sử dụng cùng bí mật yếu và không có thực thể nào trong hai thực thể đó biết trước được giá trị của khóa bí mật dùng chung.

2) **Thỏa thuận khóa được xác thực bằng mật khẩu tăng cường**: thiết lập một hoặc nhiều khóa bí mật giữa hai thực thể A và B, trong đó A dùng bí mật yếu và B có dữ liệu xác minh bắt nguồn từ hàm một chiều của bí mật yếu mà A đang dùng. Trong cơ chế thỏa thuận khóa được xác thực bằng mật khẩu tăng cường, các khóa bí mật dùng chung là kết quả của việc trao đổi dữ liệu giữa hai thực thể, tại đó khóa bí mật dùng chung được thiết lập với điều kiện hai thực thể sử dụng cùng bí mật yếu và các dữ liệu đã xác minh tương ứng và không có thực thể nào trong hai thực thể biết trước được giá trị của khóa bí mật dùng chung.

   CHÚ THÍCH Kiểu cơ chế thỏa thuận khóa này không có khả năng bảo vệ bí mật yếu của thực thể A khỏi sự thăm dò từ thực thể B, nhưng lại tăng phí tổn cho kẻ tấn công muốn biết bí mật yếu của A từ B. Do đó, kiểu cơ chế này thường được dùng cho giao tiếp giữa máy trạm (A) và máy chủ (B).

3) **Lấy lại khóa được xác thực bằng mật khẩu**: thiết lập một hoặc nhiều khóa bí mật cho một thực thể (A) liên kết với một thực thể khác (B), trong đó A dùng bí mật yếu và B sử dụng bí mật mạnh liên kết với bí mật yếu của A. Trong cơ chế lấy lại khóa được xác thực bằng mật khẩu,

các khóa bí mật có thể lấy lại được bởi A (không nhất thiết phải nhận từ B), là kết quả của việc trao đổi dữ liệu giữa hai thực thể. Trong đó khóa bí mật được thiết lập với điều kiện hai thực thể cùng dùng bí mật yếu và liên kết với bí mật mạnh. Tuy nhiên, mặc dù bí mật mạnh của B liên kết với bí mật yếu của A, thì bí mật mạnh (bản thân nó) không chứa thông tin đầy đủ cho phép xác định phép bí mật yếu hoặc khóa bí mật thiết lập trong cơ chế.

CHÚ THÍCH Kiểu cơ chế lấy lại khóa này được sử dụng khi A không có bộ lưu trữ an toàn cho bí mật mạnh và yêu cầu sự hỗ trợ từ B để lấy lại bí mật mạnh. Kiểu cơ chế này thường được sử dụng cho giao tiếp giữa máy trạm (A) và máy chủ (B).

Tiêu chuẩn này không bao gồm các vấn đề chủ chốt khác như:

- Quản lý vòng đời của bí mật yếu, bí mật mạnh và khóa bí mật đã thiết lập.
- Các cơ chế để lưu trữ, nén, xóa, hủy,... bí mật yếu, bí mật mạnh và các khóa bí mật đã thiết lập.

CHÚ THÍCH Các khóa được tạo hoặc lấy lại thông qua việc sử dụng bí mật yếu không thể an toàn hơn khi dựa vào tổng số các bí mật yếu mà chúng có. Vì vậy các cơ chế quy định trong tiêu chuẩn này được khuyến khích sử dụng trong môi trường an ninh thấp.

2 Tài liệu viện dẫn

Các tài liệu viện dẫn dưới đây rất cần thiết cho việc áp dụng tiêu chuẩn này. Đối với các tài liệu ghi năm công bố thì áp dụng phiên bản được nêu. Đối với các tài liệu không ghi năm công bố thì áp dụng phiên bản mới nhất, bao gồm cả các sửa đổi.

TCVN 7817 – 1:2007, Công nghệ thông tin – Kỹ thuật mật mã – Quản lý khoá – Phần 1:Khung tổng quát.

ISO/IEC 10118 – 3:2004, Information technology — Security techniques — Hash–functions — Part 3: Dedicated hash–functions. (*Công nghệ thông tin – Kỹ thuật an ninh – Hàm băm – Phần 3: Các hàm băm dành riêng*).

3 Thuật ngữ và định nghĩa

Tiêu chuẩn này sử dụng các định nghĩa sau:

3.1

Thỏa thuận khóa được xác thực bằng mật khẩu tăng cường (augmented password–authenticated key agreement)

Thỏa thuận khóa được xác thực bằng mật khẩu tăng cường trong đó thực thể A sử dụng mật khẩu dựa trên bí mật yếu và thực thể B xác minh dữ liệu nhận được từ hàm một chiều của bí mật yếu mà A đang dùng để thỏa thuận và xác thực một hoặc nhiều khóa bí mật dùng chung.

3.2

Thỏa thuận khóa được xác thực bằng mật khẩu cân bằng (balanced password–authenticated key agreement)

Thỏa thuận khóa được xác thực bằng mật khẩu cân bằng trong đó thực thể A và B cùng chia sẻ một bí mật yếu dựa trên mật khẩu chung để thỏa thuận và xác thực một hoặc nhiều khóa bí mật dùng chung.

3.3

Tấn công vét cạn (brute–force attack)

Kiểu tấn công vào hệ thống mật mã sử dụng phép vét cạn tập các khóa, mật khẩu hoặc dữ liệu khác.

3.4

Hàm băm kháng xung đột (collision–resistant hash–function)

Hàm băm thỏa mãn các đặc tính sau: không thể tính toán để tìm ra hai đầu vào khác nhau mà ánh xạ tới cùng một đầu ra.

CHÚ THÍCH Khả năng tính toán phụ thuộc vào môi trường và yêu cầu bảo mật cụ thể.

[ISO/IEC 10118–1:2000]

3.5

Tấn công theo từ điển (*trên hệ thống mật khẩu*) (dictionary attack (*on a password – based system*))

Tấn công vào hệ thống mật mã sử dụng phép tìm kiếm danh mục mật khẩu.

CHÚ THÍCH Tấn công theo từ điển trên hệ thống mật khẩu sử dụng danh sách các giá trị mật khẩu đã lưu hoặc danh sách các từ ngữ được lấy trong từ điển ngôn ngữ tự nhiên.

3.6

Tham số miền (domain parameter)

Mục dữ liệu chung và được nhận biết bởi hoặc có thể truy cập tới tất cả các thực thể trong miền.

CHÚ THÍCH Tập tham số miền có thể chứa các mục dữ liệu như thẻ định danh hàm băm, độ dài thẻ băm, độ dài của phần thông điệp phản hồi, tham số trường hữu hạn, tham số đường cong elliptic, hoặc các tham số khác quy định cho các vấn đề an ninh của tên miền.

[ISO/IEC 9796–3:2000]

3.7

Xác thực khóa tường minh từ A tới B (explicit key authentication from A to B)

Đảm bảo cho thực thể B rằng A là thực thể duy nhất khác B có quyền sở hữu khóa đúng.

CHÚ THÍCH Xác thực khóa không tường minh từ A tới B và xác nhận khóa từ A tới B kết hợp với nhau suy ra xác thực khóa tường minh từ A tới B.

[TCVN 7817–3:2007]

3.8

Hàm băm (hash function)

Hàm ánh xạ các chuỗi bit thành các chuỗi bit có độ dài cố định, thỏa mãn hai tính chất sau đây:

- Không có khả năng tính toán được đầu vào từ một đầu ra cho trước;
- Không có khả năng tính toán được đầu vào nào khác đầu vào cho trước mà lại có chung đầu ra.

CHÚ THÍCH Khả năng tính toán phụ thuộc vào môi trường và yêu cầu bảo mật cụ thể.

[ISO/IEC 10118–1:2000]

3.9

Mật khẩu băm (hashed password)

Kết quả của việc áp dụng hàm băm cho mật khẩu.

3.10

Xác thực khóa không tường minh từ A tới B (implicit key authentication from A to B)

Đảm bảo cho thực thể B rằng A là thực thể duy nhất khác B có khả năng có quyền sở hữu khóa đúng.

[TCVN 7817–3:2007]

3.11

Khóa (key)

Chuỗi biểu tượng điều khiển các thao tác của phép biến đổi mật mã (ví dụ như mã hóa, giải mã, tính toán hàm kiểm tra mật mã hoặc xác minh chữ ký).

[TCVN 7817–3:2007]

3.12

Thỏa thuận khóa (key agreement)

Quá trình thiết lập khóa bí mật dùng chung giữa các thực thể theo cách mà không thực thể nào có thể xác định trước giá trị của khoá đó.

[TCVN 7817–1:2007]

3.13

Xác nhận khóa từ A tới B (key confirmation from A to B)

Đảm bảo đối với thực thể B rằng thực thể A là có quyền sở hữu khóa đúng.

[TCVN 7817–3:2007]

3.14

Kiểm soát khoá (key control)

Khả năng lựa chọn khoá hoặc các tham số sử dụng trong việc tính toán khoá.

[TCVN 7817–1:2007]

3.15

Hàm dẫn xuất khóa (key derivation function)

Hàm sử dụng các bí mật dùng chung hoặc các tham số đã biết khác nhau làm đầu vào và cho đầu ra là một hay nhiều bí mật dùng chung mà có thể được sử dụng làm khóa.

3.16

Thiết lập khóa (key establishment)

Quá trình tạo sẵn khoá bí mật dùng chung cho một hoặc nhiều thực thể, thiết lập khóa bao gồm thỏa thuận khóa, chuyển khóa và lấy lại khóa.

3.17

Quản lý khóa (key management)

Quản trị và sử dụng để tạo ra, đăng ký, chứng thực, hủy bỏ, phân bổ, cài đặt, lưu trữ, nén, thu hồi, dẫn xuất và phá hủy vật liệu khóa theo quy định an ninh.

[TCVN 7817–1:2007]

3.18

Lấy lại khóa (key retrieval)

Quá trình thiết lập khóa cho một hoặc nhiều thực thể (được gọi là thực thể lấy khóa) với một hoặc nhiều thực thể khác (không bắt buộc có khả năng tiếp cận khóa sau khi quá trình kết thúc) và quá trình này thường đòi hỏi việc xác thực đối với thực thể lấy lại khóa bởi các khóa kia.

3.19

Thẻ khóa (key token)

Thông điệp thiết lập khóa gửi từ thực thể này đến thực thể khác trong suốt quá trình thực hiện cơ chế thiết lập khóa.

3.20

Hàm kiểm tra thẻ khóa (key token check function)

Hàm sử dụng thẻ khóa và các tham số đã được công khai là đầu vào và đầu ra là một giá trị Boolean trong suốt quá trình thực hiện cơ chế thiết lập khóa.

3.21

Hệ số thẻ khóa (Key token factor)

Giá trị được giữ bí mật và có thể được sử dụng kết hợp với bí mật yếu để tạo ra thẻ khóa.

3.22

Hàm tạo thẻ khóa (key token generation function)

Hàm tận dụng hệ số thẻ khóa và các số tham số khác làm đầu vào, đầu ra là thẻ khóa trong suốt quá trình thiết lập khóa.

3.23

Xác thực khóa hai chiều (mutual key authentication)

Đảm bảo cho mỗi trong hai thực thể rằng chỉ có thực thể kia có khả năng có quyền sở hữu khóa đúng.

3.24

Hàm một chiều (one–way function)

Hàm với đặc tính là dễ dàng tính toán đầu ra với đầu vào cho trước nhưng không có khả năng tính toán đầu vào với đầu ra cho trước.

[TCVN 7817–3:2007]

3.25

Mật khẩu (password)

Chuỗi các từ, cụm từ, số hoặc ký tự bí mật để sử dụng xác thực thực thể và là bí mật yếu dễ nhớ.

3.26

Thỏa thuận khóa được xác thực bằng mật khẩu (password–authenticated key agreement)

Quá trình thiết lập một hoặc nhiều khóa bí mật dùng chung giữa hai thực thể sử dụng phần thông tin dựa trên mật khẩu chia sẻ (có nghĩa là cả hai cùng chia sẻ mật khẩu hoặc một thực thể có mật khẩu và thực thể còn lại có dữ liệu xác minh mật khẩu) và không thực thể nào có thể đoán trước được giá trị các khóa bí mật dùng chung.

3.27

Lấy lại khóa được xác thực bằng mật khẩu (password–authenticated key retrieval)

Quá trình lấy lại khóa trong đó thực thể A dùng bí mật yếu bằng mật khẩu và thực thể B sở hữu bí mật mạnh liên kết với bí mật yếu từ A. Hai thực thể đó sử dụng các bí mật riêng của chúng, thỏa thuận khóa bí mật mà khóa đó có thể lấy lại lại từ A (không nhất thiết phải nhận từ B).

3.28

Thẻ khóa được làm rối bằng mật khẩu (password–entangled key token)

Thẻ khóa được sinh ra từ bí mật yếu và hệ số thẻ khóa.

3.29

Dữ liệu xác minh mật khẩu (password verification data)

Dữ liệu mà được dùng để xác minh các thông tin về thực thể của một mật khẩu cụ thể.

3.30

Hàm dẫn xuất phần tử ngẫu nhiên (random element derivation function)

Hàm sử dụng mật khẩu và các tham số khác làm đầu vào và đầu ra là phần tử ngẫu nhiên.

3.31

Salt

Biến ngẫu nhiên được thêm vào như là đầu vào thứ hai cho hàm mã hóa hoặc hàm một chiều, để tạo ra dữ liệu xác minh mật khẩu.

3.32

Bí mật (secret)

Giá trị mà chỉ có thực thể tạo ra giá trị đó biết.

3.33

Hàm dẫn xuất giá trị bí mật (Secret value derivation function)

Hàm sử dụng hệ số thẻ khóa, thẻ khóa và vài tham số khác làm đầu vào và đầu ra là một giá trị bí mật (để tính một hoặc nhiều khóa bí mật).

3.34

Khóa bí mật (secret key)

Khóa được sử dụng trong kỹ thuật mã hóa đối xứng bởi một tập các thực thể xác định.

3.35

Bí mật mạnh (Strong secret)

Bí mật với mức độ thích hợp của entropy để việc dò soát kiểu vét cạn bí mật này là không khả thi, thậm chí với các dữ liệu cho trước cho phép đoán trước chính xác bí mật để phân biệt với các giả định sai.

CHÚ THÍCH Điều này là có thể, ví dụ: bằng cách sử dụng ngẫu nhiên các bí mật từ một tập các giá trị có độ lớn thích hợp với phân bố xác suất đều đặn.

3.36

Bí mật yếu (Weak secret)

Bí mật mà con người có thể ghi nhớ dễ dàng, đặc trưng của phương pháp này là entropy của bí mật bị giới hạn, vì vậy việc dò soát kiểu vét cạn bí mật này là khả thi, các dữ liệu cho trước cho phép phán đoán chính xác bí mật để phân biệt với các giả định sai.

4 Ký hiệu và từ viết tắt

a_1, a_2	Hệ số đường cong elliptic
A,B	Định danh phân biệt của thực thể
b, b_i	bit (tức là 0 hoặc 1)
BS2I	Hàm biến đổi chuỗi bit thành số nguyên
c	Số nguyên xác định trong $1 \le c \le q - 1$
C, C_{DL}, C_{EC}	Hàm tạo ra thẻ khóa dựa trên mật khẩu và hệ số thẻ khóa
D, D_{DL}, D_{EC}	Hàm tạo ra thẻ khóa chỉ dựa trên duy nhất hệ số thẻ khóa
E	Đường cong elliptic được xác định bởi hai hệ số đường cong elliptic a_1 và a_2
F(q)	Trường hữu hạn có q phần tử
FE2I	Hàm biến đổi phần tử trường thành số nguyên
FE2OS	Hàm biến đổi phần tử trường thành chuỗi octet
g, g_1, g_a, g_b	Các phần tử có bậc nhân r trong F(q)
G, G_a, G_b	Các điểm có bậc r trên E thuộc trường F(q)
g_{q-1}	Phần tử có bậc nhân là q–1 trong F(q)
$GE2OS_X$	Hàm có khả năng chuyển đổi phần tử nhóm thành chuỗi octet, khi phần tử nhóm đó là một điểm trên E, thì hàm này sẽ chuyển tọa độ x của điểm đó thành chuỗi octet mà bỏ qua tọa độ y.
H	Hàm băm lấy chuỗi octet làm đầu vào và cho đầu ra là chuỗi bit. Ví dụ: hàm băm được định nghĩa trong ISO/IEC 10118–3
$h(x, L_K)$	Hàm băm lấy chuỗi octet x và số nguyên L_K (chỉ ra độ dài theo bit của đầu ra) làm đầu vào và cho đầu ra là chuỗi bit có độ dài L_K. Ví dụ: hàm băm được định nghĩa trong ISO/IEC 10118–3.
I2FE	Hàm biến đổi số nguyên thành phần tử trường
I2OS	Hàm biến đổi số nguyên thành chuỗi octet
I2P	Hàm biến đổi số nguyên thành điểm trên đường cong E
J, J_{DL}, J_{EC}	Hàm tạo ra phần tử xác minh mật khẩu từ mật khẩu

k	Phần phụ đại số hoặc là tỉ số (q–1)/r trong các tham số miền DL hoặc tỉ số #E/r trong các tham số miền EC
K	Hàm dẫn xuất khóa từ giá trị bí mật và tham số dẫn xuất khóa
$K_1, K_2, ..$	Các khóa bí mật được thiết lập trong một cơ chế thiết lập khóa
L_K	Độ dài (tính bằng bit) khóa bí mật đã được thiết lập
m	Một số nguyên
M_i	Một octet được biểu diễn giá trị mã hex từ 00 đến FF
mod	Phép toán đồng dư, trong đó y = a mod b là định nghĩa cho số nguyên duy nhất y trong $0 \le y < b$ và (a –y) là một bội số nguyên của b
n	Một số nguyên
o_A, o_A', o_B, o_B'	Chuỗi bit, dùng để xác định một quá trình xác nhận khóa
OS2I	Hàm biến đổi chuỗi octet thành số nguyên
p, p_i	Số nguyên tố lẻ
$P_1, P_2, ...$	Chuỗi octet tham số dẫn xuất khóa
q	Số phần tử trong trường hữu hạn F(q). Trong cấu hình EC, q hoặc là p hoặc là 2^m đối với các số nguyên $m \ge 1$. Trong cấu hình DL, q chính là p. CHÚ THÍCH Tiêu chuẩn này chỉ áp dụng cho trường nguyên tố hoặc trường nhị phân trong cấu hình EC và chỉ trường nguyên tố trong cấu hình DL, bởi vì các trường hợp này được sử dụng rộng rãi và chúng có các đặc tính bảo mật cao được nghiên cứu kỹ lưỡng.
r	Bậc của nhóm sẽ được sử dụng (là số nguyên tố hoặc chia hết cho (q–1) trong cấu hình DL và chia hết cho #E trong cấu hình EC)
$R, R_{1DL}, R_{1EC}, R_{2DL}, R_{2EC}$	Các hàm dẫn xuất phần tử ngẫu nhiên từ mật khẩu.
s_A, s_B	Các hệ số thẻ khóa của thực thể A và B tương ứng với các thẻ khóa w_A và w_B CHÚ THÍCH Các hệ số thẻ khóa nên được tạo một cách ngẫu nhiên từ một tập các chọn lựa làm cho tối đa hóa độ khó trong việc lấy lại hệ số thẻ khóa bởi các phương pháp tìm kiếm xung đột. Các phương pháp tạo số ngẫu nhiên được quy định trong ISO/IEC 18031.
T	Hàm kiểm tra tính hợp lệ của thẻ khóa
$V, V_A, V_B, V_{ADL}, V_{AEC}, V_{BDL}, V_{BEC}$	Hàm tạo các giá trị bí mật

w_A, w_B	Các thẻ khóa hoặc các thẻ khóa đã được làm rối bằng mật khẩu của các thực thể A và B, tương ứng với các hệ số thẻ khóa s_A và s_B; chúng là các số nguyên trong cấu hình DL và các điểm trong cấu hình EC.
$[x] \times Y$	Toán tử nhân trong cấu hình EC có đầu vào là một số nguyên x và một điểm Y trên đường cong E và tạo ra một điểm Z trên đường cong E, trong đó $Z = [x] \times Y = Y + Y + \ldots + Y$, với x–1 lần số hạng Y nếu x là dương. Các toán tử thỏa mãn $[0] \times Y = 0_E$ (điểm ở vô cực) và $[-x] \times Y = [x] \times (-Y)$.
z	Giá trị bí mật được sử dụng để dẫn xuất khóa, đây là một số nguyên trong cấu hình DL và là một điểm trong cấu hình EC
$\{\beta_{m-1}, \beta_{m-2}, \ldots, \beta_0\}$	Các phần tử của trường $F(s^m)$, trong đó s hoặc là p hoặc là 2 và β_i là một số nguyên thỏa mãn $0 \le \beta_i \le s-1$
π	Một chuỗi octet dựa trên mật khẩu thường bắt nguồn từ một mật khẩu hoặc một mật khẩu băm, các thẻ định danh của một hoặc nhiều thực thể, thẻ định danh của mỗi phiên giao tiếp nếu nhiều hơn một phiên được thực hiện đồng thời, tùy chọn các giá trị *Salt* và các dữ liệu khác. CHÚ THÍCH Yêu cầu cần thiết là đưa một hay nhiều định danh thực thể và một định danh của phiên liên lạc vào giá trị π, điều này tránh được cơ chế thiết lập khóa bị hư hỏng bởi kiểu tấn công chia sẻ khóa chưa biết được trình bày trong [TC05].
$\#E$	Số điểm trên đường cong elliptic E
||	Toán tử ghép nối được áp dụng cho chuỗi octet
0_E	Điểm vô cực trên đường cong elliptic E

5 Các yêu cầu

Giả thiết rằng các thực thể nhận biết được định danh của các thực thể khác. Điều này có thể thực hiện bằng cách lồng định danh vào trong thông tin trao đổi giữa hai thực thể, hoặc có thể nhận ra ngay từ ngữ cảnh sử dụng cơ chế.

Giả thiết rằng các thực thể nhận biết được tập các tham số miền chung được sử dụng để tính toán ra tập các hàm trong cơ chế thiết lập khóa. Mỗi cơ chế có thể sử dụng với một trong hai tập tham số miền khác nhau, tùy thuộc vào việc cơ chế vận hành trên nhóm giá trị nhân trong dãy F(q) hoặc trên nhóm phần tử cộng trong đường cong elliptic xác định trên dãy F(q). Trong trường hợp đầu, cơ chế được cho là hoạt động trong cấu hình DL (lôgarit rời rạc) và trong trường hợp thứ hai, cơ chế được cho là hoạt động trong cấu hình EC (đường cong elliptic).

CHÚ THÍCH Điều quan trọng cơ bản để thực hiện chính xác thao tác trong các cơ chế là bất kỳ tham số miền nào đều được giữ chính xác bởi mỗi bên tham gia. Khi bất kỳ bên tham gia nào vô tình hay cố ý sử dụng tham số miền bị lỗi có thể tạo lỗi trong các cơ chế, kéo theo bên thứ ba khám phá ra được khóa bí mật đã thiết lập.

Hai tập tham số miền là:

Tập các tham số miền DL bao gồm:

F(q) – Trường hữu hạn trên q phần tử.

q – Số phần tử trong F(q), đây là một số nguyên tố.

r – Bậc của nhóm phần tử sẽ được sử dụng thuộc trường hữu hạn, đây là ước số nguyên của q–1.

g – Phần tử nhân bậc r thuộc F(q) (g gọi là phần tử sinh của nhóm con gồm r phần tử trong F(q)).

g_{q-1} – Phần tử nhân bậc q–1 thuộc F(q).

CHÚ THÍCH Phương pháp tạo g_{q-1} có thể tìm thấy trong chương 4 của [MvV96] và [Ka86].

k – Giá trị bằng (q–1)/r, được gọi là phần phụ đại số, thỏa mãn $k = 2\ p_1 p_2 \ldots p_t$ trong đó số nguyên tố $p_i > r$ và i = 1, 2, ..., t. (Khi t = 0 thì k = 2).

Tập các tham số miền EC bao gồm:

F(q) – Trường hữu hạn trên q phần tử.

q – Số phần tử trong F(q), có giá trị

- p nếu là số nguyên tố lẻ; hoặc,
- 2^m đối với số nguyên dương $m \geq 1$.

a_1, a_2 – Hai hệ số đường cong, các phần tử F(q) được xác định trên đường cong elliptic E.

E – Đường cong elliptic xác định bởi hệ số đường cong elliptic a_1, a_2. Và được xác định bởi một trong hai phương trình sau cùng với điểm 0_E gọi là điểm vô cực:

- $Y^2 = X^3 + a_1X + a_2$ trên trường F(p),
- $Y^2 + XY = X^3 + a_1X^2 + a_2$ trên trường $F(2^m)$,

#E – Số các điểm trên E.

r – Bậc của nhóm mong muốn, đó là số nguyên chia hết cho #E.

G – Một điểm trên đường cong bậc r (G gọi là phần tử sinh của nhóm con gồm r điểm trên E).

k – Giá trị bằng #E/r, được gọi là phần phụ đại số, thỏa mãn $k = 2^n\ p_1 p_2 \ldots p_t$ trong đó n = {0, 1, 2} và số nguyên tố $p_i > r$ và i = 1, 2, ..., t. (Khi t = 0 thì $k = 2^n$).

Khi các thực thể sử dụng một cơ chế cụ thể trong cấu hình EC, giả định rằng các thực thể nhận biết được hình thức biểu diễn điểm, nghĩa là điểm được biểu diễn dưới dạng nén hoặc không nén hoặc dạng mẫu lai. Các quy định kỹ thuật về biểu diễn điểm xem thêm trong ISO/IEC 18033–2.

Trong đặc tả kỹ thuật các cơ chế của tiêu chuẩn này, phương pháp tạo số ngẫu nhiên tuân theo ISO/IEC 18031 và phương pháp tạo số nguyên tố tuân theo ISO/IEC 18032.

Cũng giả thiết rằng các thực thể nhận biết được một hàm băm H chung, ví dụ: hàm băm riêng được quy định trong ISO/IEC 10118–3.

6 Thỏa thuận khóa được xác thực bằng mật khẩu

Điều khoản này quy định ba cơ chế thỏa thuận khóa được xác thực bằng mật khẩu. Cơ chế đầu tiên quy định trong điều 6.1 là cơ chế thỏa thuận khóa được xác thực bằng mật khẩu cân bằng đòi hỏi hai thực thể dùng chung một bí mật yếu. Các cơ chế thứ hai và thứ ba, quy định tại điều 6.2 và 6.3 là các cơ chế thỏa thuận khóa được xác thực bằng mật khẩu tăng cường đòi hỏi một trong hai thực thể có dữ liệu xác minh bí mật yếu từ thực thể khác.

Tất cả ba cơ chế thỏa thuận khóa được xác thực bằng mật khẩu đều được thực hiện tiếp theo sau quá trình khởi tạo và quá trình thiết lập khóa.

Quá trình khởi tạo: Hai thực thể tham gia đồng ý sử dụng một tập các tham số miền hợp lệ, một tập các tham số dẫn xuất khóa và một tập các hàm, tất cả đều được công khai. Hai thực thể cũng đồng ý sử dụng hoặc bí mật yếu dựa trên mật khẩu chia sẻ mà chỉ họ biết hoặc thông tin dựa trên mật khẩu chia sẻ, nghĩa là một thực thể có bí mật yếu dựa trên mật khẩu và thực thể còn lại có dữ liệu xác minh mật khẩu tương ứng.

Quá trình thiết lập khóa:

1) *Tạo và trao đổi thẻ khóa*: từng thực thể trong hai thực thể liên quan lựa chọn ngẫu nhiên một hoặc nhiều hệ số thẻ khóa liên kết với các tham số miền, tạo ra các thẻ khóa tương ứng mà có thể liên kết được với mật khẩu hoặc dữ liệu xác minh mật khẩu (thẻ này được gọi là thẻ được làm rối bằng mật khẩu) và sau đó tạo ra các thẻ khóa cho thực thể còn lại.

2) *Kiểm tra tính hợp lệ của thẻ khóa*: tùy thuộc vào các thao tác trong quá trình sản xuất thẻ khóa ở bước 1, từng thực thể liên quan lựa chọn một phương pháp thích hợp để xác nhận là đã nhận được các thẻ khóa dựa trên tham số miền. Nếu có bất kỳ xác nhận nào thất bại, đầu ra sẽ là "không hợp lệ" và quá trình dừng lại.

3) *Thu được khóa bí mật dùng chung*: từng thực thể trong hai thực thể liên quan áp dụng một hàm dẫn xuất giá trị bí mật nhất định để sở hữu riêng hệ số thẻ khóa, các thẻ khóa và/hoặc mật khẩu chia sẻ hoặc dữ liệu xác minh mật khẩu của thực thể khác nhằm tạo ra được giá trị chia sẻ bí mật. Mỗi thực thể áp dụng thêm một hàm dẫn xuất khóa với giá trị chia sẻ bí mật và các tham số dẫn xuất khóa để thu được một hoặc nhiều khóa bí mật dùng chung.

4) *Kiểm tra xác minh khóa*: hai thực thể liên quan sử dụng thiết lập khóa bí mật dùng chung với các bước trên thì nhận được khóa cho mỗi bên. Bước này là tùy chọn trong cơ chế 1 nhưng bắt buộc trong cơ chế 2 và 3.

6.1 Cơ chế thỏa thuận khóa 1

Đây là cơ chế thỏa thuận khóa được thiết kế theo thỏa thuận khóa được xác thực bằng mật khẩu cân bằng, trong đó thiết lập một hay nhiều khóa bí mật dùng chung giữa hai thực thể A và B với kiểm soát khóa kết nối và chia sẻ đầu tiên chuỗi octet dựa trên mật khẩu π. Cơ chế này cung cấp xác thực khóa không tường minh hai chiều và tùy chọn xác nhận khóa tường minh hai chiều.

Cơ chế này làm việc được trên cả hai cấu hình DL và EC.

CHÚ THÍCH Cơ chế này dựa trên [Jab96] và cơ chế được gọi là {DL,EC}BPKAS–SPEKE trong [IEEEP1363.2].

6.1.1 Tham số được chia sẻ trước

Thỏa thuận khóa giữa hai thực thể A và B diễn ra trong môi trường chứa các tham số sau đây:

- Chuỗi octet dựa trên mật khẩu π dùng chung.
- Tập các tham số miền hợp lệ (tham số miền DL hoặc tham số miền EC) quy định trong Điều 5.
- Hàm dẫn xuất phần tử ngẫu nhiên, R.
- Hàm tạo thẻ khóa, D.
- Hàm kiểm tra thẻ khóa, T.
- Hàm dẫn xuất giá trị bí mật, V.
- Hàm dẫn xuất khóa, K.
- Giá trị Boolean b cho biết liệu phép nhân phần phụ đại số có được yêu cầu hay không. Nếu b=1, thì phép nhân phần phụ đại số được yêu cầu, nếu khác thì không được yêu cầu.
- Một hoặc nhiều chuỗi octet tham số dẫn xuất khóa {P_1, P_2, ...}, trong đó A và B phải cùng chấp nhận sử dụng giá trị P_i .
- Độ dài của khóa bí mật dùng chung, L_K.

CHÚ THÍCH Phép nhân phần phụ đại số được sử dụng để ánh xạ các thẻ khóa nhận được thành một phần tử nhóm hợp lệ, tức là một phần tử trong nhóm con đã chọn bậc r. Trường hợp b=0 chỉ được dùng trong các cơ chế mà trong đó được đảm bảo rằng nhận được thẻ khóa là phần tử nhóm hợp lệ. Xem chi tiết hơn về phép nhân phần phụ đại số trong [ISO/IEC 15946–3:2002].

6.1.2 Các hàm

6.1.2.1 Hàm dẫn xuất phần tử ngẫu nhiên R

Hàm dẫn xuất phần tử ngẫu nhiên R lấy chuỗi octet x làm đầu vào và tạo ra đầu ra là phần tử nhóm được chọn R(x). Cơ chế thỏa thuận khóa 1 có thể sử dụng bất kỳ một trong bốn hàm R sau: R_{1DL}, R_{1EC}, R_{2DL} và R_{2EC}.

- R_{1DL} tương ứng khi cơ chế sử dụng tham số miền DL, tức là hoạt động trên nhóm phần tử nhân xác định trên F(q). Với tham số miền DL cho trước (bao gồm k và q) và chuỗi octet x làm đầu vào thì ta có R_{1DL}:

$$R_{1DL}(x) = (BS2I(H(x)))^k \bmod q.$$

- R_{1EC} tương ứng khi cơ chế sử dụng tham số miền EC, tức là hoạt động trên nhóm phần tử cộng thuộc đường cong elliptic xác định trên F(q). Với tham số miền EC cho trước (bao gồm k) và chuỗi octet x làm đầu vào thì ta có R_{1EC}:

$$R_{1EC}(x) = [k] \times I2P(BS2I(H(x))).$$

- R_{2DL} tương ứng khi cơ chế sử dụng tham số miền DL, tức là hoạt động trên nhóm phần tử nhân xác định trên F(q). Với tham số miền DL cho trước (bao gồm q), hai phần tử ngẫu nhiên thuộc nhóm con bậc r trong F(q) là g_a và g_b và chuỗi octet x làm đầu vào thì ta có R_{2DL}:

$$R_{2DL}(x) = g_a * g_b^{\,BS2I(H(x))} \bmod q.$$

- R_{2EC} tương ứng khi cơ chế sử dụng tham số miền EC, tức là hoạt động trên nhóm phần tử cộng thuộc đường cong elliptic xác định trên F(q). Với tham số miền EC cho trước, hai phần tử ngẫu nhiên thuộc nhóm con bậc r trên E là G_a và G_b và chuỗi octet x làm đầu vào thì ta có R_{2EC}:

$$R_{2EC}(x) = G_a + [BS2I(H(x))] \times G_b.$$

Hàm BS2I (chuyển chuỗi bit thành số nguyên) và I2P (chuyển số nguyên thành điểm) được miêu tả trong Phụ lục A.

CHÚ THÍCH 1 Bốn lựa chọn cho hàm R có các đặc điểm hiệu suất và giả thiết bảo mật khác nhau. Về hiệu suất, R_2 được sử dụng khi k >> r, nhưng khi sử dụng phần phụ đại số k nhỏ, R_1 lại nhanh hơn R_2.

CHÚ THÍCH 2 Khuyến cáo rằng nếu kết quả của $R_{1DL}(x)$ hoặc $R_{2DL}(x)$ là 1, hoặc nếu kết quả của $R_{1EC}(x)$ hoặc $R_{2EC}(x)$ là 0_E, thì đầu ra là "không hợp lệ" và quá trình dừng lại. Căn cứ vào đặc tính ngẫu nhiên của hàm băm H, xác suất sảy ra trường hợp này là rất ít. Tuy nhiên không điểm yếu bảo mật nào bị phát hiện, bởi vì nếu hàm R cho đầu ra không ngừng có giá trị 1 trong cấu hình DL hoặc điểm 0_E trong cấu hình EC, thì giao thức sẽ hủy bỏ khi chạy hàm kiểm tra thẻ khóa T.

6.1.2.2 Hàm tạo thẻ khóa D

Hàm tạo thẻ khóa D lấy số nguyên x và phần tử nhóm y làm đầu vào và tạo ra đầu ra là nhóm phần tử D(x, y). Cơ chế thiết lập khóa 1 có thể được sử dụng một trong các hàm D sau: D_{DL} và D_{EC} :

- D_{DL} tương ứng với cơ chế sử dụng các tham số miền DL, tức là hoạt động trên nhóm phần tử nhân xác định trên F(q). Cho trước tham số miền DL (bao gồm q) và hai đầu vào x từ {1,…,r–1} và số nguyên y là đầu ra hàm R, trong đó D_{DL} tính như sau:

$$D_{DL}(x, y) = y^x \bmod q.$$

- D_{EC} tương ứng với cơ chế sử dụng các tham số miền EC, tức là hoạt động trên nhóm phần tử cộng thuộc đường cong elliptic xác định trên F(q). Cho trước tham số miền EC và hai đầu vào x từ {1, …, r – 1} và điểm Y là đầu ra hàm R, trong đó D_{EC} tính như sau:

$$D_{EC}(x, Y) = [x] \times Y.$$

6.1.2.3 Hàm kiểm tra thẻ khóa T

Hàm kiểm tra thẻ khóa T lấy phần tử nhóm x làm đầu vào và tạo ra đầu ra là giá trị Boolean T(x). Cơ chế thỏa thuận khóa 1 sử dụng một trong các hàm T sau, T_{DL} và T_{EC} :

- T_{DL} tương ứng với cơ chế sử dụng các tham số miền DL, tức là hoạt động trên nhóm phần tử nhân thuộc F(q). Cho trước tham số miền DL (bao gồm q) và chuỗi dữ liệu x, trong đó T_{DL} được xác định như sau:
  - Nếu x không biểu diễn số nguyên, $T_{DL}(x) = 0$,
  - Nếu $x \leq 1$, $T_{DL}(x) = 0$,
  - Nếu $x \geq q - 1$, $T_{DL}(x) = 0$.
  - Trường hợp còn lại, $T_{DL}(x) = 1$.
- T_{EC} tương ứng với cơ chế sử dụng tham số miền EC, tức là hoạt động trên nhóm phần tử cộng thuộc đường cong elliptic xác định trên F(q). Với tham số miền EC (bao gồm 0_E) cho trước, giá trị $n \in \{0, 1, 2\}$, cho $k = 2^n p_1 p_2 \ldots p_t$ và chuỗi dữ liệu X, T_{EC} được xác định như sau:
  - Nếu X không biểu diễn một điểm trên E, $T_{EC}(X) = 0$,
  - Nếu $[2^n] \times X = 0_E$, $T_{EC}(X) = 0$.
  - Trường hợp còn lại, $T_{EC}(X) = 1$.

6.1.2.4 Hàm dẫn xuất giá trị bí mật V

Hàm dẫn xuất giá trị bí mật V hoạt động trên số nguyên x và phần tử nhóm đã chọn y với giá trị Boolean b làm đầu vào và tạo ra đầu ra là phần tử nhóm V(x, y, b). Cơ chế thỏa thuận khóa 1 có thể chọn một trong các hàm V sau, V_{DL} và V_{EC}:

- V_{DL} tương ứng với cơ chế sử dụng tham số miền DL, tức là hoạt động trên nhóm phần tử nhân trên F(*q*). Cho trước tham số miền DL (bao gồm k và *q*) với ba đầu vào: x từ {1, …, r – 1}, y từ {2, …, *q* – 2} và b từ {0, 1}, V_{DL} được xác định như sau:

$$V_{DL}(x, y, b) = y^{x*k^b} \bmod q\ .$$

- V_{EC} tương ứng với cơ chế sử dụng tham số miền EC, tức là hoạt động trên nhóm phần tử cộng thuộc đường cong elliptic trên F(q). Cho trước tham số miền EC (bao gồm k) và ba đầu vào: x từ {1, …, r – 1}, điểm $Y \neq 0_E$ và b từ {0, 1}, V_{EC} được xác định như sau:

$$V_{EC}(x, Y, b) = [k^b * x] \times Y.$$

6.1.2.5 Hàm dẫn xuất khoá K

Hàm dẫn xuất khóa K thao tác trên chuỗi octet x, độ dài (tính bằng bit) L_K của đầu ra hàm K và chuỗi octet tham số dẫn xuất khóa P từ $\{P_1, P_2, ...\}$ làm đầu vào, tạo ra đầu ra là chuỗi bit $K(x, P, L_K)$. Cơ chế thỏa thuận khóa 1 sử dụng hàm một chiều là hàm K:

$$K(x, P, L_K) = h(x \| P, L_K).$$

CHÚ THÍCH 1 Việc chuyển đổi đầu ra cho hàm băm được quy định trong ISO/IEC 10118 – 3 là mã băm H với chiều dài bit cho trước. Xem ISO/IEC 10118 – 3 để biết thêm chi tiết.

CHÚ THÍCH 2 Giá trị của L_K phụ thuộc vào ứng dụng sử dụng khóa dẫn xuất. Nếu đầu ra của hàm dẫn xuất khóa K là một khóa mã hóa đối xứng, thì L_K là giá trị của chiều dài khóa của cơ chế mã hóa đối xứng cụ thể.

6.1.3 Thao tác thỏa thuận khóa

Cơ chế này yêu cầu cả hai A và B chấp nhận một dãy bốn bước, từ A1 – A4 và B1 – B4 (các bước tương ứng với A và B). Các bước tùy chọn là A3, A4, B3 và B4.

Xây dựng thẻ khóa (A1)

A tiến hành các bước sau:

- Tính $g_1 = R(\pi)$ làm cơ sở cho thẻ khóa.
- Chọn một số nguyên s_A ngẫu nhiên từ $\{1, ..., r – 1\}$ làm hệ số thẻ khóa.
- Tính $w_A = D(s_A , g_1)$ là thẻ khóa.
- Chuyển w_A cho B.

Xây dựng thẻ khóa (B1)

B tiến hành các bước sau:

- Tính $g_1 = R(\pi)$ làm cơ sở cho thẻ khóa.
- Chọn một số nguyên s_B ngẫu nhiên từ $\{1, ..., r – 1\}$ là hệ số thẻ khóa.
- Tính $w_B = D(s_B , g_1)$ là thẻ khóa.
- Chuyển w_B cho A.

Thu được khóa bí mật dùng chung (A2)

A tiến hành các bước sau:

- Nhận w_B từ B.
- Kiểm tra tính hợp lệ của w_B bằng $T(w_B)$: Nếu $T(w_B) = 0$, đầu ra sẽ "không hợp lệ" và quá trình dừng lại; nếu không thì:
- Tính $z = V(s_A, w_B, b)$ là giá trị bí mật dùng chung.
- Tính $K_i = K(GE2OS_X(z), P_i, L_K)$ cho từng tham số dẫn xuất khóa P_i là khóa bí mật dùng chung.

Thu được khóa bí mật dùng chung (B2)

B tiến hành các bước sau:

- Nhận w_A từ A.
- Kiểm tra tính hợp lệ của w_A bằng $T(w_A)$: Nếu $T(w_A) = 0$, đầu ra sẽ "không hợp lệ" và quá trình dừng lại; nếu không thì:
- Tính $z = V(s_B, w_A, b)$ là giá trị bí mật dùng chung.
- Tính $K_i = K(GE2OS_X(z), P_i, L_K)$ cho từng tham số dẫn xuất khóa P_i là khóa bí mật dùng chung.

CHÚ THÍCH Cơ chế không quy định thứ tự nào giữa A1 và B1 hoặc A2 và B2, ngoài ra theo tính logic yêu cầu cần tính toán giá trị trước khi sử dụng giá trị đó nên A1 và B1 phải xuất hiện trước A2 và B2.

Xác minh khóa (A3 và B3) *(tùy chọn)*

A thực hiện các bước sau (A3):

- Tính $o_A = H(hex(03) \| GE2OS_X(w_A) \| GE2OS_X(w_B) \| GE2OS_X(z) \| GE2OSX(g_1))$ và
- Chuyển o_A cho B.

B thực hiện các bước sau (B3):

- Nhận o_A từ A.
- Tính $o'_A = H(hex(03) \| GE2OS_X(w_A) \| GE2OS_X(w_B) \| GE2OS_X(z) \| GE2OSX(g_1))$ và
- Nếu $o_A \neq o'_A$ thì đầu ra là "không hợp lệ", quá trình dừng lại.

Xác minh khóa (B4 và A4) *(tùy chọn)*

B thực hiện các bước sau (B4):

- Tính $o_B = H(hex(04) \| GE2OS_X(w_A) \| GE2OS_X(w_B) \| GE2OS_X(z) \| GE2OSX(g_1))$ và
- Chuyển o_B cho A.

A thực hiện các bước sau (A4):

- Nhận o_B từ B.
- Tính $o'_B = H(hex(04) \| GE2OS_X(w_A) \| GE2OS_X(w_B) \| GE2OS_X(z) \| GE2OSX(g_1))$ và
- Nếu $o_B \neq o'_B$ thì đầu ra là "không hợp lệ", quá trình dừng lại.

CHÚ THÍCH Các thực thể A và B có thể tự do chọn A3 và B3 hoặc A4 và B4. Tuy nhiên B3 phải xuất hiện sau A3 hoặc A4 phải xuất hiện sau B4.

Hàm GE2OSX (chuyển nhóm phần tử thành chuỗi octet) được mô tả trong Phụ lục A.

CHÚ THÍCH Phần tử nhóm trong cơ chế này là điểm trên đường cong E trong cấu hình EC, hoặc một số nguyên trong dãy [1, q – 1] thuộc cấu hình DL.

6.2 Cơ chế thỏa thuận khóa 2

Đây là cơ chế thỏa thuận khóa được thiết kế theo thỏa thuận khóa được xác thực bằng mật khẩu tăng cường, trong đó thiết lập một hay nhiều khóa bí mật dùng chung giữa hai thực thể A và B với việc kiểm soát khóa kết nối. Trong cơ chế này, A có chuỗi octet dựa trên mật khẩu π và B có dữ liệu xác minh mật khẩu v tương ứng với π. Cơ chế này cung cấp thỏa thuận khóa tường minh một chiều và tùy chọn xác nhận khóa hai chiều.

Cơ chế này làm việc trên cấu hình DL.

CHÚ THÍCH 1 Trong ứng dụng sử dụng thỏa thuận khóa được xác thực bằng mật khẩu tăng cường, A đóng vai trò máy trạm và B đóng vai trò máy chủ.

CHÚ THÍCH 2 Cơ chế này dựa trên [Wu02] và cơ chế được gọi là DLAPKAS–SRP6 trong [IEEEP1363.2].

6.2.1 Tham số được chia sẻ trước

Thỏa thuận khóa giữa hai thực thể A và B diễn ra trong một môi trường bao gồm các tham số sau:

- Tập các tham số miền DL, bao gồm g_{q-1} và q, xác định trong Điều 5.
- Chuỗi octet dựa trên mật khẩu π, sử dụng bởi A.
- Phần tử xác minh mật khẩu $v = J(\pi)$ sử dụng bởi B, trong đó J là hàm dẫn xuất phần tử xác minh mật khẩu.
- Hàm tạo thẻ khóa D, sử dụng bởi A.
- Hàm tạo thẻ khóa được làm rối bằng mật khẩu C, sử dụng bởi B.
- Hai hàm dẫn xuất giá trị bí mật, V_A và V_B tương ứng với từng thực thể.
- Hàm dẫn xuất khóa, K.
- Một hoặc nhiều chuỗi octet tham số dẫn xuất khóa {P_1, P_2, ...}, trong đó A và B phải thỏa thuận cùng sử dụng giá trị P_i.
- Số nguyên c sao cho $c = (BS2I(H(I2OS(g_{q-1}) \,||\, I2OS(q)))$ mod q.
- Độ dài của khóa bí mật dùng chung L_K.

Hàm BS2I (chuyển chuỗi bit thành số nguyên) và I2OS (chuyển số nguyên thành chuỗi octet) xác định trong Phụ lục A.

6.2.2 Các hàm

6.2.2.1 Hàm dẫn xuất phần tử xác minh mật khẩu J

Hàm dẫn suất phần tử xác minh mật khẩu J hoạt động trên chuỗi octet dựa trên mật khẩu π làm đầu vào và tạo ra đầu ra là số nguyên $J(\pi)$. Cơ chế thỏa thuận khóa 2 sử dụng hàm J như sau:

- Cho trước tham số miền DL (bao gồm cả g_{q-1} và q) và chuỗi octet dựa trên mật khẩu π làm đầu vào thì *J* được tính như sau:

$$J(\pi) = g_{q-1}^{BS2I(H(\pi))} \bmod q.$$

Hàm BS2I (chuyển chuỗi bit thành số nguyên) được xác định trong phụ lục A.

6.2.2.2 Hàm tạo thẻ khóa D

Hàm tạo thẻ khóa D lấy số nguyên x từ {1, …, q – 2} làm đầu vào và tạo ra đầu ra là số nguyên D(x). Cơ chế thỏa thuận khóa 2 sử dụng hàm D như sau:

- Cho trước tham số miền DL (bao gồm g_{q-1} và q) và đầu vào x từ {1, …, q – 2} thì D được tính như sau:

$$D(x) = g_{q-1}^{x} \bmod q.$$

6.2.2.3 Hàm tạo thẻ khóa được làm rối bằng mật khẩu C

Hàm tạo thẻ khóa được làm rối bằng mật khẩu *C* hoạt động trên ba đầu vào: số nguyên *c*, số nguyên x từ {1, …, q – 2} và đầu ra của hàm dẫn xuất phần tử xác minh mật khẩu $v = J(\pi)$ và tạo ra đầu ra là số nguyên *C(x, v, c)*. Cơ chế thỏa thuận khóa 2 sử dụng hàm *C* như sau:

- Cho trước tham số miền DL (bao gồm cả g_{q-1} và q) và ba đầu vào: số nguyên *c*, *x* từ {1,…,q – 2} và đầu ra *v* của hàm *J*, khi đó *C* được tính như sau:

$$C(x, v, c) = v * c + g_{q-1}^{x} \bmod q.$$

6.2.2.4 Hàm dẫn xuất giá trị mật V_A và V_B

1) Hàm dẫn xuất giá trị mật V_A hoạt động trên sáu đầu vào: số nguyên c, chuỗi octet dựa trên mật khẩu π, số nguyên x_A từ {1,…,q–2}, số nguyên *v* là đầu ra của hàm dẫn xuất phần tử xác minh mật khẩu *J*, số nguyên y_A là đầu ra của hàm tạo thẻ khóa *D*, số nguyên y_B là đầu ra của hàm tạo thẻ khóa được làm rối bằng mật khẩu *C* và tạo ra đầu ra là số nguyên $V_A(c, \pi, x_A, v, y_A, y_B)$.
2) Hàm dẫn xuất giá trị mật V_B hoạt động trên bốn đầu vào: số nguyên x_B từ {1,…,q–2}, số nguyên *v* là đầu ra của hàm dẫn xuất phần tử xác minh mật khẩu *J*, số nguyên y_A là đầu ra của hàm tạo thẻ khóa *D*, số nguyên y_B là đầu ra của hàm tạo thẻ khóa được làm rối bằng mật khẩu *C* và tạo ra đầu ra là số nguyên $V_B(x_B, v, y_A, y_B)$.
3) V_A và V_B thỏa mãn điều kiện $V_A(c, \pi, x_A, v, y_A, y_B) = V_B(x_B, v, y_A, y_B)$.

Cơ chế thỏa mãn khóa 2 có thể sử dụng V_A và V_B theo sau:

1) Cho trước tham số miền DL (bao gồm g_{q-1} và q), số nguyên *c*, chuỗi octet dựa trên mật khẩu π, số nguyên x_A từ {1, …, q – 2}, đầu ra *v* của hàm *J*, đầu ra y_A của hàm *D*, đầu ra y_B của hàm *C*, khi đó V_A được tính như sau:

- Tính $u_1 = BS2I(H(\pi))$,
- Tính $u_2 = BS2I(H(I2OS(y_A) \| I2OS(y_B)))$ và
- Tính $V_A(c, \pi, x_A, v, y_A, y_B) = (y_B - v * c)^{(x_A + u_1 * u_2)} \bmod q$.

2) Cho trước tham số miền DL (bao gồm g_{q-1} và q), số nguyên x_B từ {1, ..., q – 2}, đầu ra v của hàm J, đầu ra y_A của hàm D, đầu ra y_B của hàm C thì khi đó V_B được tính như sau:
   - Tính $u = BS2I(H(I2OS(y_A) \| I2OS(y_B)))$ và
   - Tính $V_B(x_B, v, y_A, y_B) = (y_A * v^u)^{x_B} \bmod q$.

Hàm I2OS (chuyển số nguyên thành chuỗi octet) và BS2I (chuyển chuỗi bit thành số nguyên) được miêu tả trong Phụ lục A.

6.2.2.5 Hàm dẫn xuất khóa K

Hàm dẫn xuất khóa K tương tự như đã xác định trong Điều 6.1.2.5.

6.2.3 Thao tác thỏa thuận khóa

Cơ chế này yêu cầu cả hai A và B chấp nhận dãy bốn bước, từ A1 – A4 và B1 – B4 (các bước tương ứng với A và B). Các bước tùy chọn là A4 và B4.

Xây dựng thẻ khóa (A1)

A tiến hành các bước sau:

- Chọn số nguyên s_A ngẫu nhiên từ {1, ..., q – 2} là hệ số thẻ khóa.
- Tính $w_A = D(s_A)$ là thẻ khóa.
- Chuyển w_A cho B.

Xây dựng thẻ khóa được làm rối bằng mật khẩu (B1)

B tiến hành các bước sau:

- Nhận w_A từ A.
- Kiểm tra nếu $1 < w_A < q-1$, nếu không đúng thì đầu ra là "không hợp lệ", quá trình dừng lại, nếu đúng thì tiếp tục,
- Chọn số nguyên s_A ngẫu nhiên từ {1, ..., q – 2} là hệ số thẻ khóa.
- Tính $w_B = C(s_B, v, c)$ là thẻ khóa được làm rối bằng mật khẩu.
- Chuyển w_A cho A.

Thu được khóa bí mật dùng chung (A2)

A tiến hành các bước sau:

- Nhận w_B từ B.
- Kiểm tra nếu $1< w_B <q-1$, nếu không đúng thì đầu ra là "không hợp lệ", quá trình dừng lại, nếu đúng thì tiếp tục,
- Tính $z = V_A(c, \pi, s_A, v, w_A, w_B)$ là giá trị thỏa thuận bí mật.
- Tính $K_i = K(I2OS(z), P_i, L_K)$ là khóa bí mật dùng chung cho từng tham số dẫn xuất khóa P_i.

Thu được khóa bí mật dùng chung (B2)

B tiến hành các bước sau:

- Tính $z = V_B(s_B, v, w_A, w_B)$ là giá trị thỏa thuận bí mật.
- Tính $K_i = K(I2OS(z), P_i, L_K)$ là khóa bí mật dùng chung cho từng tham số dẫn xuất khóa P_i.

Xác minh khóa (A3 và B3) *(bắt buộc)*

A thực hiện các bước sau (A3):

- Tính $o_A = H(hex(04) \| I2OS(w_A) \| I2OS(w_B) \| I2OS(z) \| I2OS(v))$ và
- Chuyển o_A cho B.

B thực hiện các bước sau (B3):

- Nhận o_A từ A.
- Tính $o'_A = H(hex(04) \| I2OS(w_A) \| I2OS(w_B) \| I2OS(z) \| I2OS(v))$ và
- Nếu $o_A \neq o'_A$ thì đầu ra là "không hợp lệ", quá trình dừng lại.

Xác minh khóa (B4 và A4) *(tùy chọn)*

B thực hiện các bước sau (B4):

- Tính $o_B = H(hex(03) \| I2OS(w_A) \| I2OS(w_B) \| I2OS(z) \| I2OS(v))$ và
- Chuyển o_B cho A.

A thực hiện các bước sau (A4):

- Nhận o_B từ B.
- Tính $o'_B = H(hex(03) \| I2OS(w_A) \| I2OS(w_B) \| I2OS(z) \| I2OS(v))$ và
- Nếu $o_B \neq o'_B$ thì đầu ra là "không hợp lệ", quá trình dừng lại.

Hàm I2OS (chuyển số nguyên thành chuỗi octet) được miêu tả trong Phụ lục A.

CHÚ THÍCH Thực thể B phải xác minh bằng chứng của thực thể A về khóa thỏa thuận trước khi nhận bất kỳ thông tin nào thu được từ khóa thỏa thuận. Do đó A3/B3 phải thực hiện trước A4/B4.

6.3 Cơ chế thỏa thuận khóa 3

Đây là cơ chế thỏa thuận khóa được thiết kế theo thỏa thuận khóa được xác thực bằng mật khẩu tăng cường, trong đó thiết lập một hay nhiều khóa bí mật dùng chung giữa hai thực thể A và B với kiểm soát khóa kết nối. Trong cơ chế này, A có chuỗi octet dựa trên mật khẩu π và B có dữ liệu xác minh mật khẩu v tương ứng với π. Cơ chế này cung cấp thỏa thuận khóa tường minh một chiều và tùy chọn xác nhận khóa hai chiều.

Cơ chế này làm việc được trên cấu hình DL và cấu hình EC.

CHÚ THÍCH 1 Trong các ứng dụng sử dụng thỏa thuận khóa được xác thực bằng mật khẩu tăng cường, thực thể A phải đóng vai trò máy trạm và thực thể B phải đóng vai trò máy chủ.

CHÚ THÍCH 2 Cơ chế này dựa trên [Kw00] và [Kw03] và cơ chế được gọi là {DL, EC}APKAS–AMP trong [IEEEP1363.2].

6.3.1 Tham số được chia sẻ trước

Thỏa thuận khóa giữa hai thực thể A và B diễn ra trong một môi trường chứa các tham số sau:

- Chuỗi octet dựa trên mật khẩu π được sử dụng bởi A.
- Tập các tham số miền hợp lệ (tham số miền DL hoặc tham số miền EC) xác định trong Điều 5.
- Phần tử xác minh mật khẩu $v = J(\pi)$ sử dụng bởi B, trong đó J là hàm dẫn xuất phần tử xác minh mật khẩu.
- Hàm tạo thẻ khóa D, sử dụng bởi A.
- Hàm tạo thẻ khóa được làm rối bằng mật khẩu C, được sử dụng bởi B.
- Hàm kiểm tra thẻ khóa T.
- Hai hàm dẫn xuất giá trị bí mật V_A và V_B tương ứng với từng thực thể.
- Hàm dẫn xuất khóa K.
- Một hoặc nhiều chuỗi octet tham số dẫn xuất khóa $\{P_1, P_2, ...\}$, trong đó A và B phải cùng chấp nhận sử dụng giá trị P_i.
- Độ dài của khóa bí mật dùng chung, L_K.

6.3.2 Các hàm

6.3.2.1 Hàm dẫn xuất phần tử xác minh mật khẩu J

Hàm dẫn xuất phần tử xác minh mật khẩu J lấy chuỗi octet dựa trên mật khẩu π làm đầu vào và tạo ra đầu ra là phần tử nhóm được chọn $J(\pi)$ xác định trên $F(q)$. Cơ chế thỏa thuận khóa 3 có thể sử dụng một trong các hàm J_{DL} và J_{EC} sau:

- J_{DL} thỏa mãn sao cho cơ chế sử dụng các tham số miền DL, tức là hoạt động trên nhóm phần tử nhân trên F(q). Cho trước các tham số miền DL (bao gồm g và q) và chuỗi octet dựa trên mật khẩu π thì J_{DL} được tính như sau:

$$J_{DL}(\pi) = g^{BS2I(H(\pi))} \bmod q.$$

- J_{EC} thỏa mãn sao cho cơ chế sử dụng các tham số miền EC, tức là hoạt động trên nhóm phần tử cộng thuộc đường cong elliptic trên F(q). Cho trước các tham số miền EC (bao gồm G) và chuỗi octet dựa trên mật khẩu π thì J_{EC} được tính như sau:

$$J_{EC}(\pi) = [BS2I(H(\pi))] \times G.$$

Hàm BS2I (chuyển chuỗi bit thành số nguyên) được xác định trong Phụ lục A.

6.3.2.2 Hàm tạo thẻ khóa D

Hàm tạo thẻ khóa D hoạt động trên số nguyên x từ {1, …, r – 1} làm đầu vào và tạo ra đầu ra là nhóm phần tử *D(x)* đã chọn. Cơ chế thỏa thuận khóa 3 có thể sử dụng một trong hai hàm D_{DL} và D_{EC}:

- D_{DL} tương ứng với cơ chế sử dụng tham số miền *DL*, tức là hoạt động trên nhóm phần tử nhân xác định trên F(q). Cho trước tham số miền *DL* (bao gồm g và q) và đầu vào x từ {1,…, r – 1}, trong đó D_{DL} tính như sau:

$$D_{DL}(x) = g^x \bmod q.$$

- D_{EC} tương ứng với cơ chế sử dụng tham số miền *EC*, tức là hoạt động trên nhóm phần tử cộng thuộc đường cong elliptic xác định trên F(q). Cho trước tham số miền *EC* (bao gồm G) và đầu vào x từ {1, …, r – 1}, trong đó D_{EC} tính như sau:

$$D_{EC}(x) = [x] \times G.$$

6.3.2.3 Hàm tạo thẻ khóa được làm rối bằng mật khẩu C

Hàm tạo thẻ khóa được làm rối bằng mật khẩu *C* hoạt động trên ba đầu vào: số nguyên x từ {1,…, r–1} và đầu ra *v* (hoặc *V*) của hàm *J* và đầu ra *y* (hoặc *Y*) của hàm *D* và tạo ra đầu ra là phần tử nhóm được chọn *C(x, v, y)*. Cơ chế thỏa thuận khóa 3 có thể sử dụng một trong hai hàm C_{DL} và C_{EC}:

- C_{DL} tương ứng với cơ chế sử dụng tham số miền *DL*, tức là hoạt động trên nhóm phần tử nhân xác định trên *F(q)*. Cho trước tham số miền *DL* (bao gồm q) và ba đầu vào bao gồm: số nguyên x từ {1, …, r – 1} và đầu ra *v* của hàm *J*, đầu ra *y* của hàm *D*, khi đó C_{DL} được tính như sau:
  - Tính $e = BS2I(H(I2OS(1) \,||\, GE2OS_X(y)))$,
  - Tính $C_{DL}(x, v, y) = (v * y^e)^x \bmod q$,
  - Kiểm tra $C_{DL}(x, v, y)$ bằng 1 hoặc (q – 1) thì đầu ra là "không hợp lệ", quá trình sẽ dừng lại. Nếu không thì $C_{DL}(x, v, y)$ là giá trị cần.

- C_{EC} tương ứng với cơ chế sử dụng tham số miền *EC*, tức là hoạt động trên nhóm phần tử cộng thuộc đường cong elliptic xác định trên F(q). Cho trước tham số miền *EC* và ba đầu vào: số nguyên x từ {1, …, r – 1} và đầu ra *V* của hàm *J*, đầu ra *Y* của hàm *D*, khi đó C_{EC} tính như sau:
  - Tính *e* = *BS2I(H(I2OS(1) || GE2OS$_X$ (Y)))*,
  - Tính $C_{EC}(x,V,Y)=[x]\times(V+[e]\times Y)$,
  - Kiểm tra nếu $[4]\times C_{EC}(x,V,Y)=0_E$ thì đầu ra là "không hợp lệ", quá trình sẽ dừng lại. Nếu không thì $C_{EC}(x,V,Y)$ là giá trị cần.

Hàm BS2I (chuyển chuỗi bit thành số nguyên) và hàm I2OS (chuyển số nguyên thành chuỗi octet) và hàm GE2OS$_X$ (chuyển phần tử nhóm thành chuỗi octet) được miêu tả trong Phụ lục A.

6.3.2.4 Hàm kiểm tra thẻ khóa T

Hàm kiểm tra thẻ khóa T tương tự đã xác định trong Điều 6.1.2.3.

6.3.2.5 Hàm dẫn xuất giá trị mật V_A và V_B

1) Hàm dẫn xuất giá trị mật V_A hoạt động trên bốn đầu vào: chuỗi octet dựa trên mật khẩu π, số nguyên x_A từ {1, …, r–1}, y_A (hoặc Y_A) của hàm *D*, số y_B (hoặc Y_B) của hàm *C* và tạo ra đầu ra là phần tử nhóm $V_A(\pi, x_A, y_A, y_B)$.
2) Hàm dẫn xuất giá trị mật V_B hoạt động trên ba đầu vào: số nguyên x_B từ {1, …, r–1}, y_A (hoặc Y_A) của hàm *D*, số y_B (hoặc Y_B) của hàm *C* và tạo ra đầu ra là phần tử nhóm $V_B(x_B, y_A, y_B)$.
3) V_A và V_B thỏa mãn điều kiện $V_A(\pi, x_A, y_A, y_B) = V_B(x_B, y_A, y_B)$.

Cơ chế thỏa thuận khóa 3 có thể sử dụng một trong hai hàm V_A là: V_{ADL} và V_{AEC} và sử dụng một trong hai hàm V_B là: V_{BDL} và V_{BEC} như sau:

1. V_{ADL} thỏa mãn cơ chế sử dụng tham số miền *DL*, tức là hoạt động trên nhóm nhân trên F(q). Cho trước tham số miền DL (bao gồm r và q), chuỗi octet dựa trên mật khẩu π, số nguyên x_A từ {1,…, r–1}, đầu ra y_A từ {2,…, q–2}, đầu ra y_B từ {2, …, q–2} thì khi đó V_{ADL} được tính như sau:
   - Tính *e* = *BS2I(H(I2OS(1) || GE2OS$_X$ (y_A)))*,
   - Tính *d* = *BS2I(H(I2OS(2) || GE2OS$_X$ (y_A) || GE2OS$_X$ (y_B)))*,
   - Tính $u = (x_A + d)/(x_A * e + BS2I(H(\pi))) \bmod r$,
   - Tính $V_{ADL}(\pi, x_A, y_A, y_B) = y_B^u \bmod q$.
   - Đầu ra $V_{ADL}(\pi, x_A, y_A, y_B)$.
2. V_{BDL} thỏa mãn cơ chế sử dụng tham số miền *DL*, tức là hoạt động trên nhóm nhân trên F(q). Cho trước tham số miền *DL* (bao gồm g và q), số nguyên x_B từ {1, …, r – 1}, đầu ra y_A từ {2, …, q – 2}, đầu ra y_B từ {2, …, q – 2} thì khi đó V_{BDL} được tính như sau:

- Tính $d = BS2I(H(I2OS(2) \| GE2OS_X(y_A) \| GE2OS_X(y_B)))$
- Tính $V_{BDL}(x_B, y_A, y_B) = (y_A * g^d)^{x_B} \bmod q$,
- Đầu ra $V_{BDL}(x_B, y_A, y_B)$

3. V_{AEC} thỏa mãn cơ chế sử dụng tham số miền *EC*, tức là hoạt động trên nhóm cộng thuộc đường cong elliptic trên *F(q)*. Cho trước tham số miền *EC* (bao gồm r), chuỗi octet dựa trên mật khẩu π, số nguyên x_A từ {1, …, r – 1}, điểm Y_A ($\neq 0_E$) trên E, đầu ra Y_B ($\neq 0_E$) trên *E* thì khi đó V_{AEC} được tính như sau:
   - Tính $e = BS2I(H(I2OS(1) \| GE2OS_X(Y_A)))$,
   - Tính $d = BS2I(H(I2OS(2) \| GE2OS_X(Y_A) \| GE2OS_X(Y_B)))$,
   - Tính $u = (x_A + d)/(x_A * e + BS2I(H(\pi))) \bmod r$,
   - Tính $V_{AEC}(\pi, x_A, Y_A, Y_B) = [u] \times Y_B$,
   - Đầu ra $V_{AEC}(\pi, x_A, Y_A, Y_B)$.
4. V_{BEC} thỏa mãn cơ chế sử dụng tham số miền *EC*, tức là hoạt động trên nhóm cộng thuộc đường cong elliptic trên *F(q)*. Cho trước tham số miền *EC* (bao gồm cả *G*), số nguyên x_B từ {1,…, r – 1}, điểm Y_A ($\neq 0_E$) trên E, đầu ra Y_B ($\neq 0_E$) trên *E* thì khi đó V_{BEC} được tính như sau:
   - Tính $d = BS2I(H(I2OS(2) \| GE2OS_X(Y_A) \| GE2OS_X(Y_B)))$
   - Tính $V_{BEC}(x_B, Y_A, Y_B) = [x_B] \times (Y_A + [d] \times G)$,
   - Đầu ra $V_{BEC}(x_B\ Y_A, Y_B)$.

Hàm I2OS (chuyển số nguyên thành chuỗi octet) và BS2I (chuyển chuỗi bit thành số nguyên) và hàm $GE2OS_X$ (chuyển phần tử nhóm thành chuỗi octet) được miêu tả trong Phụ lục A.

6.3.2.6 Hàm dẫn xuất khóa K

Hàm dẫn xuất khóa K tương tự đã xác định trong Điều 6.1.2.5.

6.3.3 Thao tác thỏa thuận khóa

Cơ chế này yêu cầu cả hai A và B chấp nhận dãy bốn bước, từ A1 – A4 và B1 – B4 (các bước tương ứng với A và B). Các bước tùy chọn là A4 và B4.

Xây dựng thẻ khóa (A1)

A tiến hành các bước sau:

- Chọn số nguyên s_A ngẫu nhiên từ {1, …, r – 1} là hệ số thẻ khóa.
- Tính $w_A = D(s_A)$ là thẻ khóa.
- Chuyển w_A cho B.

Xây dựng thẻ khóa được làm rối bằng mật khẩu (B1)

B tiến hành các bước sau:

- Nhận w_A từ A.
- Kiểm tra tính hợp lệ của w_A sử dụng $T(w_A)$: nếu $T(w_A) = 0$, thì đầu ra là "không hợp lệ", quá trình dừng lại, nếu khác thì tiếp tục,
- Chọn số nguyên s_B ngẫu nhiên từ {1, …, r – 1} làm hệ số thẻ khóa.
- Tính $w_B = C(s_B, v, w_A)$ là thẻ khóa được làm rối bằng mật khẩu (nếu đầu ra hàm C là "không hợp lệ" thì quay lại tìm s_B khác với giá trị thích hợp), sau đó.
- Chuyển w_B cho A.

Thu được khóa bí mật dùng chung (A2)

A tiến hành các bước sau:

- Nhận w_B từ B.
- Kiểm tra tính hợp lệ của w_B sử dụng $T(w_B)$: nếu $T(w_B) = 0$, thì đầu ra là "không hợp lệ", quá trình dừng lại, nếu khác thì tiếp tục,
- Tính $z = V_A(\pi, s_A, w_A, w_B)$ là giá trị thỏa thuận bí mật.
- Tính $K_i = K(GE2OS_X(z), P_i, L_K)$ là khóa bí mật dùng chung cho từng tham số dẫn xuất khóa P_i.

Thu được khóa bí mật dùng chung (B2)

B tiến hành các bước sau:

- Tính $z = V_B(s_B, w_A, w_B)$ là giá trị thỏa thuận bí mật.
- Tính $K_i = K(GE2OS_X(z), P_i, L_K)$ là khóa bí mật dùng chung cho từng tham số dẫn xuất khóa P_i.

Xác minh khóa (A3 và B3) *(bắt buộc)*

A thực hiện các bước sau (A3):

- Tính $o_A = H(I2OS(4) \,||\, GE2OS_X(w_A) \,||\, GE2OS_X(w_B) \,||\, GE2OS_X(z))$ và
- Chuyển o_A cho B.

B thực hiện các bước sau (B3):

- Nhận o_A từ A.
- Tính $o'_A = H(I2OS(4) \,||\, GE2OS_X(w_A) \,||\, GE2OS_X(w_B) \,||\, GE2OS_X(z))$ và
- Nếu $o_A \neq o'_A$ thì đầu ra là "không hợp lệ", quá trình dừng lại.

Xác minh khóa (B4 và A4) *(tùy chọn)*

B thực hiện các bước sau (B4):

- Tính $o_B = H(I2OS(3) \,||\, GE2OS_X(w_A) \,||\, GE2OS_X(w_B) \,||\, GE2OS_X(z))$ và
- Chuyển o_B cho A.

A thực hiện các bước sau (A4):

- Nhận o_B từ B.
- Tính $o'_B = H(I2OS(3) \,||\, GE2OS_X(w_A) \,||\, GE2OS_X(w_B) \,||\, GE2OS_X(z))$ và
- Nếu $o_B \neq o'_B$ thì đầu ra là "không hợp lệ", quá trình dừng lại.

Hàm $GE2OS_X$ (chuyển phần tử nhóm thành chuỗi octet) xác định trong Phụ lục A.

CHÚ THÍCH 1 Phần tử nhóm trong cơ chế này là điểm trên đường cong E trong cấu hình EC, hoặc là số nguyên thuộc dãy $[1, q - 1]$ trong cấu hình DL.

CHÚ THÍCH 2 Thực thể B phải xác minh bằng chứng của thực thể A về khóa thỏa thuận trước khi nhận bất kỳ thông tin nào thu được từ khóa thỏa thuận. Do đó A3/B3 phải thực hiện trước B4/A4.

CHÚ THÍCH 3 – Dựa trên kiểu tấn công phân tích Pohlig–Hellman, giá trị bí mật thấp nhất một hoặc hai bit của B là s_B có thể bị nhận ra bởi kẻ tấn công, khi k được chia cho 2 hoặc 4.

7 Lấy lại khóa được xác thực bằng mật khẩu

Điều này quy định cơ chế lấy lại khóa được xác thực bằng mật khẩu. Trong cơ chế, thực thể A có bí mật yếu dẫn xuất từ mật khẩu và thực thể còn lại B có bí mật mạnh liên kết với bí mật yếu của A. Sử dụng các bí mật tương ứng của mình, hai thực thể thương lượng một khóa bí mật, có thể lấy lại từ A nhưng không nhất thiết phải nhận từ B.

Kết quả của quá trình này là A có được giá trị của khóa bí mật được dẫn xuất từ bí mật yếu của A và bí mật mạnh của B. Thực thể B không cần biết bí mật của A hoặc khóa bí mật nhận được. Bí mật của B liên kết với bí mật của A, nhưng (trong chính bản thân nó) không chứa đủ thông tin để cho phép xác định bí mật của A hoặc khóa bí mật đã thiết lập, ngay cả với tấn công vét cạn.

CHÚ THÍCH Trong ứng dụng sử dụng lấy lại khóa được xác thực bằng mật khẩu, A đóng vai trò máy khách và B đóng vai trò máy chủ.

Thao tác lấy lại khóa được xác thực bằng mật khẩu được thực hiện tiếp theo sau quá trình khởi tạo và quá trình thiết lập khóa

Quá trình khởi tạo: Hai thực thể tham gia đồng ý sử dụng tập các tham số miền hợp lệ và tập các hàm, tất cả đều được công khai. Thực thể A thiết lập một bí mật yếu dựa trên mật khẩu và thực thể còn lại B thiết lập một bí mật mạnh liên kết với bí mật yếu của A.

Quá trình thiết lập khóa:

1) *Tạo và trao đổi thẻ khóa*: Thực thể A lựa chọn một hệ số thẻ khóa, kiến tạo thẻ khóa được làm rối bằng mật khẩu và chuyển thẻ khóa đó cho thực thể B. Sau khi nhận được thẻ khóa được làm rối bằng mật khẩu từ A, thực thể B kiến tạo một thẻ khóa và chuyển nó cho A.

2) *Kiểm tra tính hợp lệ của thẻ khóa (tùy chọn)*: tùy thuộc vào các thao tác trong quá trình tạo ra thẻ khóa, từng thực thể liên quan lựa chọn một phương pháp thích hợp để chấp nhận các đóng góp khóa đã nhận và tham số miền. Nếu bất kỳ chấp nhận nào thất bại, đầu ra sẽ là "không hợp lệ" và chấm dứt quá trình.

3) *Thu được khóa bí mật dùng chung*: A áp dụng các thao tác giải mã trên chính hệ số thẻ khóa của A và thẻ khóa của B để tạo ra được giá trị bí mật và có thể áp dụng thêm một hàm dẫn xuất khóa với giá trị bí mật và một hoặc nhiều tham số dẫn xuất khóa để thu được một hoặc nhiều khóa bí mật dùng chung.

7.1 Cơ chế lấy lại khóa 1

Cơ chế lấy lại khóa này được thiết lập giống khi lấy lại khóa được xác thực bằng mật khẩu. Nó sử dụng mật khẩu được dẫn xuất từ bên khởi tạo cho các dạng thay đổi của thỏa thuận khóa Diffie–Hellma. Thực thể B xác định khóa và phân phối nó cho A.

Cơ chế làm việc trên cả cấu hình DL và cấu hình EC.

CHÚ THÍCH Cơ chế dựa trên [FK00] và cơ chế được gọi là {DL,EC}PKRS–1 trong [IEEEP1363.2].

7.1.1 Tham số được chia sẻ trước

Thao tác lấy lại khóa yêu cầu hai thực thể A và B thực hiện trong môi trường bao gồm các tham số:

- Tập các tham số miền hợp lệ (tham số miền DL hoặc tham số miền EC) xác định trong Điều 5.
- Chuỗi octet dựa trên mật khẩu π chỉ A biết.
- Số nguyên bí mật s_B trong {1, …, r – 1} được sử dụng cho hệ số thẻ khóa của B và chỉ B biết.
- Hàm dẫn xuất phần tử ngẫu nhiên R, sử dụng bởi A.
- Hàm tạo thẻ khóa D, sử dụng bởi cả A và B.
- Hàm kiểm tra thẻ khóa, T.
- Hàm dẫn xuất giá trị bí mật V, sử dụng bởi A.
- Hàm dẫn xuất khóa K, sử dụng bởi A.
- Một hoặc nhiều chuỗi octet tham số dẫn xuất khóa {P_1, P_2, ...},
- Độ dài của khóa bí mật, L_K.

7.1.2 Các hàm

7.1.2.1 Hàm dẫn xuất phần tử ngẫu nhiên R

Đây là các hàm R_{1DL} hoặc R_{1EC} đã được xác định trong Điều 6.1.2.1.

7.1.2.2 Hàm tạo thẻ khóa D

Hàm tạo thẻ khóa D giống như quy định trong Điều 6.1.2.2.

7.1.2.3 Hàm kiểm tra thẻ khóa T

Hàm kiểm tra thẻ khóa T giống như quy định trong Điều 6.1.2.3.

7.1.2.4 Hàm dẫn xuất giá trị bí mật V

Hàm dẫn xuất giá trị bí mật *V* lấy số nguyên *x* và phần tử nhóm được chọn *y* làm đầu vào và tạo ra đầu ra là phần tử nhóm *V(x, y)*. Cơ chế lấy lại khóa 1 sử dụng một trong hai hàm V sau : V_{DL} và V_{EC}.

- V_{DL} thích hợp với cơ chế sử dụng tham số miền *DL*, tức là hoạt động trên nhóm phần tử nhân xác định trên *F(q)*. Cho trước tham số miền *DL* (bao gồm *r* và *q*), hai đầu vào: *x* từ {1, … ,r–1} và *y* từ {2, … , q–2} , khi đó V_{DL} được tính như sau:

$$V_{DL}(x, y) = y^{x^{-1} \bmod r} \bmod q .$$

- V_{EC} thích hợp với cơ chế sử dụng tham số miền *EC*, tức là hoạt động trên nhóm phần tử cộng thuộc đường cong elliptic xác định trên *F(q)*. Cho trước tham số miền *EC* (bao gồm *r*), hai đầu vào: *x* từ {1, … ,r–1} và điểm $Y \neq 0_E$, khi đó V_{EC} được tính như sau:

$$V_{EC}(x, Y) = [x^{-1} \bmod r] \times Y .$$

7.1.2.5 Hàm dẫn xuất khóa K

Đây là hàm giống như hàm đã quy định trong Điều 6.1.2.5.

7.1.3 Thao tác lấy lại khóa

Cơ chế này yêu cầu thực thể A phải thực hiện hai bước A1 và A2, còn thực thể B thực hiện 1 bước B1.

Xây dựng thẻ khóa (A1)

A thực hiện các bước sau:

- Tính $g_1 = R(\pi)$ làm phần tử cơ sở của thẻ khóa,
- Chọn ngẫu nhiên s_A từ {1, …, r – 1} làm hệ số thẻ khóa,

- Thẻ khóa là $w_A = D(s_A, g_1)$
- Gửi w_A cho thực thể B.

Xây dựng thẻ khóa (B1)

B thực hiện các bước sau:

- Nhận w_A từ thực thể A,
- Kiểm tra tính hợp lệ của w_A với $T(w_A)$: nếu $T(w_A) = 0$ thì đầu ra "không hợp lệ", quá trình dừng lại, nếu khác thì tiếp tục,
- Thẻ khóa là $w_B = D(s_B, w_A)$
- Gửi w_B cho thực thể A.

Dẫn xuất khóa bí mật (A2)

A thực hiện các bước sau:

- Nhận được w_B từ thực thể B.
- Kiểm tra tính hợp lệ của w_B với $T(w_B)$: nếu $T(w_B) = 0$ thì đầu ra "không hợp lệ", quá trình dừng lại, nếu khác thì tiếp tục,
- Tính $z = V(s_A, w_B)$ là mã hóa cứng.
- Tính khóa bí mật $K_i = K(GE2OS_X(z), P_i, L_K)$ cho từng chuỗi octet tham số dẫn xuất khóa P_i trong $\{P_1, P_2, \ldots\}$.

Hàm $GE2OS_X$ (chuyển phần tử nhóm thành chuỗi octet) được xác định trong Phụ lục A.

CHÚ THÍCH 1 Phần tử nhóm trong cơ chế này là điểm thuộc đường cong E trong cấu hình EC, hoặc là số nguyên thuộc dãy [1, q – 1] trong cấu hình DL.

CHÚ THÍCH 2 Dựa trên kiểu tấn công phân tích Pohlig–Hellman, một hoặc hai bit thấp nhất của giá trị bí mật s_B của thực thể B có thể bị nhận ra bởi người tấn công, khi k chia hết cho 2 hoặc 4.

Phụ lục A

(quy định)

Hàm chuyển đổi kiểu dữ liệu

Phụ lục này quy định hàm chuyển đổi kiểu dữ liệu được sử dụng bởi các cơ chế thiết lập khóa trong tiêu chuẩn này.

A.1 I2OS & OS2I

Mục này quy định hàm *I2OS* (chuyển đổi số nguyên thành chuỗi octet) và *OS2I* (chuyển đổi chuỗi octet thành số nguyên).

Hàm *I2OS* lấy đầu vào là số nguyên không âm x và tạo ra đầu ra duy nhất là chuỗi octet $M_{l-1}, M_{l-2} \ldots M_0$ với chiều dài *l*. Trong đó $l = \lceil log256(x + 1) \rceil$ là chiều dài của x tính bằng octet. Khi đó *I2OS* được tính như sau:

1. Viết x dưới dạng biểu diễn chuỗi đơn *l* chữ số 256 bit:

$$x = x_{l-1}\, 256^{l-1} + x_{l-2}\, 256^{l-2} + \ldots + x_1\, 256 + x_0,$$

   trong đó $0 \le x_i < 256$.

2. Giả sử octet M_i có giá trị x_i với $0 \le i \le l-1$.

3. Đầu ra sẽ là chuỗi octet $M_{l-1}, M_{l-2} \ldots M_0$.

Cho ví dụ : *I2OS*(10945) = 2A C1.

Hàm *OS2I* lấy chuỗi octet $M_{l-1}, M_{l-2} \ldots M_0$ làm đầu vào và tạo ra đầu ra là một số nguyên không âm *y*. Hàm được tính như sau:

1. Lấy số nguyên y_i có giá trị của octet M_i với $0 \le i \le l-1$.

2. Tính số nguyên $y = y_{l-1}\, 256^{l-1} + y_{l-2} 256^{l-2} + \ldots + y_1\, 256 + y_0$.

3. Đầu ra là *y*.

Cho ví dụ: *OS2I*(2A C1) = 10945.

Chú ý rằng chuỗi octet có chiều dài bằng 0 (chuỗi octet rỗng) chuyển thành số nguyên 0 và ngược lại.

A.2 BS2I

Mục này xác định hàm *BS2I* (chuyển chuỗi bit thành số nguyên):

Hàm *BS2I* sử dụng chuỗi bit $b_{l-1}\, b_{l-2} \ldots b_0$ làm đầu vào và tạo ra đầu ra là số nguyên không âm. Hàm được tính như sau:

1. Lấy số nguyên y_i có giá trị của bit b_i với $0 \le i \le l-1$.
2. Tính số nguyên $y = y_{l-1} 2^{l-1} + y_{l-2} 2^{l-2} + \ldots + y_1 2 + y_0$.
3. Đầu ra là y.

Ví dụ: nếu $l = 19$, $BS2I(000\ 0010\ 1010\ 1100\ 0001) = 10945$.

Chú ý rằng chuỗi bit có độ dài bằng 0 (chuỗi bit rỗng) thì sẽ chuyển thành số nguyên 0.

A.3 FE2I & I2FE

Mục này xác định hàm *FE2I* (chuyển phần tử trường thành số nguyên) và hàm *I2FE* (chuyển số nguyên thành phần tử trường).

Lấy một phần tử thuộc trường hữu hạn $F(s^m)$ (trong đó s hoặc là p hoặc là 2) được biểu diễn bởi dãy $\{\beta_{m-1}, \beta_{m-2}, \ldots, \beta_0\}$ trong đó β_i là số nguyên thỏa mãn $0 \le \beta_i \le s - 1$.

Hàm *FE2I* sử dụng phần tử trường $\{\beta_{m-1}, \beta_{m-2}, \ldots, \beta_0\}$ làm đầu vào và đầu ra là số nguyên không âm. Hàm được xác định như sau:

1. Lấy số nguyên y_i có giá trị của β_i với $0 \le i \le m-1$.
2. Tính số nguyên $y = y_{m-1} s^{m-1} + y_{m-2} s^{m-2} + \ldots + y_1 s + y_0$.
3. Đầu ra là y.

Hàm *I2FE* lấy số nguyên không âm x làm đầu vào và đầu ra là trường phần tử $\{\beta_{m-1}, \beta_{m-2}, \ldots, \beta_0\}$. Hàm được xác định như sau:

1. Viết x dưới dạng dãy s với m chữ số và được biểu diễn như sau:

   $$x = x_{m-1} s^{m-1} + x_{m-2} s^{m-2} + \ldots + x_1 s + x_0$$

   trong đó $0 \le x_i < s$ (chú ý rằng một hoặc nhiều số đầu bằng 0 nếu $x < s^{m-1}$).
2. Lấy β_i có giá trị x_i với $0 \le i \le m - 1$.
3. Đầu ra là $\{\beta_{m-1}, \beta_{m-2}, \ldots, \beta_0\}$.

A.4 FE2OS

Mục này xác định hàm *FE2OS* (chuyển phần tử trường thành chuỗi octet).

Hàm *FE2OS* sử dụng dãy phần tử $\{\beta_{m-1}, \beta_{m-2}, \ldots, \beta_0\}$ làm đầu vào và tạo ra đầu ra là chuỗi octet y. Hàm được xác định như sau:

1. Chuyển dãy $\{\beta_{m-1}, \beta_{m-2}, \ldots, \beta_0\}$ thành số nguyên x bằng hàm *FE2I*.
2. Chuyển x thành chuỗi octet y bằng hàm *I2OS*.

A.5 GE2OS$_X$

Mục này xác định hàm *GE2OS$_X$* (chuyển phần tử nhóm thành chuỗi octet).

Hàm *GE2OS$_X$* sử dụng phần tử nhóm làm đầu vào và tạo ra đầu ra là chuỗi octet. Hàm được định nghĩa như sau:

Trong cấu hình *DL*, phần tử nhóm là phần tử trong *F(q)*. Cho *u* là một phần tử nhóm, đầu ra sẽ là *GE2OS$_X$(u)*:

1. Biểu diễn *u* dưới dạng phần tử trường.
2. Chuyển kết quả của bước (1) thành chuỗi octet bằng hàm *FE2OS*.
3. Đầu ra là kết quả của bước (2).

Trong cấu hình *EC*, phần tử nhóm là điểm trên đường cong elliptic *E*. Cho $Q = (x_Q, y_Q)$ là một điểm trên E, trong đó x_Q là hoành độ *x* của *Q* và y_Q là tung độ *y* của Q; và cả hai x_Q và y_Q đều trong *F(q)*. Với mục đích của cơ chế quy định trong tiêu chuẩn này thì hàm *GE2OS$_X$(Q)* chuyển hoành độ *x* của *Q* thành chuỗi octet và bỏ qua tung độ *y* của *Q*. Đầu ra của hàm *GE2OS$_X$(Q)* xác định như sau:

1. Biểu diễn x_Q dưới dạng phần tử trường.
2. Chuyển kết quả của bước (1) thành chuỗi octet bằng hàm *FE2OS*.
3. Đầu ra là kết quả của bước (2).

CHÚ THÍCH Phép chuyển đổi không phải là ánh xạ 1 – 1. Ví dụ: phép chuyển đổi trên sẽ liên kết các điểm đường cong elliptic *Q* và *–Q* với cùng một chuỗi octet.

A.6 I2P

Mục này xác định hàm *I2P* (chuyển số nguyên thành điểm).

Cho trước tập tham số miền *EC* là $(E, q, p, m, r, k, a_1, a_2)$, hàm *I2P* lấy số nguyên *u* làm đầu vào và tạo ra đầu ra là điểm *T* thuộc đường cong *E* trên *F(q)*, trong đó *T = I2P(u)*. Trong những quy định dưới đây, các toán tử cộng và nhân giữa các phần tử trường hữu hạn theo quy định trong ISO/IEC 15946–1.

1. Thiết lập $v = BS2I(H(I2OS(u)))$ mod *q*.
   - Nếu $v = 0$, đầu ra là "không hợp lệ", quá trình dừng lại.
2. Thiết lập $\lambda = u$ mod 2.
3. Nếu *q* là số nguyên tố $(q = p)$ và đường cong *E* là $Y^2 = X^3 + a_1X + a_2$ thuộc *F(q)*. Điểm *T* được xác định như sau:

   (a) Thiết lập $x = v$.

   (b) Tính toán trường phần tử $\alpha = x^3 + a_1x + a_2$ mod *p*.
   - Nếu $\alpha = 0$, đầu ra là "không hợp lệ" quá trình phải ngừng lại.

(c) Tìm căn bậc hai β của α mod p (tức là một số nguyên β với $0 < \beta < p$ sao cho $\beta^2 = \alpha \bmod p$) hoặc chứng minh không tồn tại căn bậc hai,

- Để kiểm tra tính tồn tại của căn bậc hai, tính $\delta = \alpha^{(p-1)/2} \bmod p$. Nếu thu được $\delta = 1$, thì β tồn tại. Ngược lại thì β không tồn tại.
- Nếu $\delta \neq 1$ ta lập lại bước 1 với $u = u + 1 \bmod p$.
- Nếu $\delta = 1$, thì tìm β.

CHÚ THÍCH Các thao tác tìm phần tử trường β sao cho $\beta^2 = \alpha \bmod p$ được quy định trong [ANSI X9.62] và [IEEE P1363].

(d) Thiết lập $y = (p - 1)^{\lambda} \times \beta$.

(e) Thiết lập điểm $T = (x, y)$ là đầu ra.

4. Nếu q là chẵn ($q = 2^m$) và đường cong E là $Y^2 + XY = X^3 + a_1X^2 + a_2$ trên $F(2^m)$. Điểm T được xác định như sau:

(a) Thiết lập $x = I2FE(v)$.

(b) Thiết lập $\alpha = x + a_1 + a_2x^{(-2)}$ trong $F(2^m)$.

(c) Tìm phần tử trường β sao cho $\beta^2 + \beta \equiv \alpha$ trong $F(2^m)$ hoặc xác minh rằng không tồn tại. Nếu không tồn tại thì cho $u = u + 1 \bmod q$ và quay lại bước 1.

CHÚ THÍCH Các thao tác xác minh tồn tại và tìm phần tử trường β sao cho $\beta^2 + \beta = \alpha$ trong $F(2^m)$ được quy định trong [ANSI X9.62] và [IEEE P1363].

(d) Thiết lập $y = (\beta + I2FE(\lambda)) \times x$.

(e) Thiết lập điểm $T = (x, y)$ làm đầu ra.

Phụ lục B

(quy định)

Mô đun ASN.1

Sau đây là các mô đun ASN.1 dành cho các cơ chế quản lý khóa được quy định trong tiêu chuẩn này.

```
KeyManagement-WeakSecrets {
   iso(1) standard(0) keyManagement(11770)
      weakSecrets(4) asn1-module(0) object-identifiers(0) }
 DEFINITIONS EXPLICIT TAGS ::=
 BEGIN
                                  -- EXPORTS All; --
                                 -- IMPORTS None; --
 OID ::= OBJECT IDENTIFIER -- Alias
                                   -- Đồng nghĩa --
 id-km-ws OID ::= {
   iso(1) standard(0) keyManagement(11770) weakSecrets(4) }
                                    -- Phép gán --
id-km-ws-kAM-1 OID ::= { id-km-ws keyAgreementMechanism-1(1) }
id-km-ws-kAM-2 OID ::= { id-km-ws keyAgreementMechanism-2(2) }
id-km-ws-kAM-3 OID ::= { id-km-ws keyAgreementMechanism-3(3) }
id-km-ws-kRM-1 OID ::= { id-km-ws keyRetrievalMechanism-1(4) }
                          -- Cơ chế thỏa thuận khóa 1 --
 SharedPassword-based ::= OCTET STRING
 randomElementDerivation-1 OID ::= {
   id-km-ws-kAM-1 randomElementDerivationFunction(1) }
 keyTokenGeneration-1 OID ::= {
   id-km-ws-kAM-1 keyTokenGenerationFunction(2) }
 keyTokenCheck-1 OID ::= {
   id-km-ws-kAM-1 keyTokenCheckFunction(3) }
 secretValueDerivation-1 OID ::= {
   id-km-ws-kAM-1 secretValueDerivationFunction(4) }
 keyDerivation-1 OID ::= {
   id-km-ws-kAM-1 keyDerivationFunction(5) }
 CofactorMultiplication ::= BOOLEAN
 desired    CofactorMultiplication ::= TRUE
notDesired CofactorMultiplication ::= FALSE
KeyDerivationParameterStrings ::=
      SEQUENCE SIZE(1..MAX) OF OCTET STRING
                          -- Cơ chế thỏa thuận khóa 2 --
 Password-based ::= OCTET STRING
 passwordVerification-2 OID ::= {
   id-km-ws-kAM-2 passwordVerificationFunction(1) }
 keyTokenGeneration-2 OID ::= {
   id-km-ws-kAM-2 keyTokenGenerationFunction(2) }
 passwordEntangledKeyTokenGeneration-2 OID ::= {
   id-km-ws-kAM-2 passwordEntangledKeyTokenGenerationFunction(3) }
 secretValueDerivation-2 OID ::= {
   id-km-ws-kAM-2 secretValueDerivationFunction(4) }
 keyDerivation-2 OID ::= {
   id-km-ws-kAM-2 keyDerivationFunction(5) }
                          -- Cơ chế thỏa thuận khóa 3 --
 passwordVerification-3 OID ::= {
   id-km-ws-kAM-3 passwordVerificationFunction(1) }
 keyTokenGeneration-3 OID ::= {
   id-km-ws-kAM-3 keyTokenGenerationFunction(2) }
 passwordEntangledKeyTokenGeneration-3 OID ::= {
   id-km-ws-kAM-3 passwordEntangledKeyTokenGenerationFunction(3) }
 keyTokenCheck-3 OID ::= {
```

```
  id-km-ws-kAM-3 keyTokenCheckFunction(4) }
secretValueDerivation-3 OID ::= {
  id-km-ws-kAM-3 secretValueDerivationFunction(5) }
keyDerivation-3 OID ::= {
  id-km-ws-kAM-3 keyDerivationFunction(6) }
                          -- Cơ chế lấy lại khóa 1 --
SecretInteger ::= INTEGER
randomElementDerivation-4 OID ::= {
  id-km-ws-kRM-1 randomElementDerivationFunction(1) }
keyTokenGeneration-4 OID ::= {
  id-km-ws-kRM-1 keyTokenGenerationFunction(2) }
keyTokenCheck-4 OID ::= {
  id-km-ws-kRM-1 keyTokenCheckFunction(3) }
secretValueDerivation-4 OID ::= {
  id-km-ws-kRM-1 secretValueDerivationFunction(4) }
keyDerivation-4 OID ::= {
  id-km-ws-kRM-1 keyDerivationFunction(5) }
                    END  -- Quản lý khóa bằng bí mật yếu --
```

Phụ lục C
(tham khảo)

Hướng dẫn lựa chọn tham số

Sau đây là hướng dẫn lựa chọn tham số cho các cơ chế quản lý khóa trong tiêu chuẩn này.

C.1 Tham số q, r và k

Như đã đề cập tại Điều 4, tiêu chuẩn này chỉ xử lý trường số nguyên tố *F(q)* hoặc trường nhị phân $F(2^m)$ trong cấu hình *EC* và chỉ xử lý trường số nguyên tố trong cấu hình *DL*, bởi vì các trường hợp này được sử dụng khá rộng rãi và các đặc tính bảo mật của chúng rất tốt.

Trong cấu hình DL, được kiến nghị rằng:

- q là số nguyên tố lớn, với độ lớn ít nhất là 1024 bit.
- r là ước số nguyên tố lớn của q–1, với độ lớn ít nhất là 160 bit.
- $k = (q–1)/r$. Để giảm độ phức tạp trong việc kiểm tra tính hợp lệ của thông điệp nhận được trong cơ chế thiết lập khóa được quy định trong tiêu chuẩn này, yêu cầu thỏa mãn $k = 2p_1 p_2 \dots p_t$, với số nguyên tố $p_i > r$, $i = 1, 2, \dots, t$. Tùy chọn $k = 2$, trong trường hợp này, $q = 2r + 1$ là số nguyên tố an toàn.

Trong cấu hình EC, được kiến nghị rằng:

- Trong trường số nguyên tố *F(q)* thì *q* là số nguyên tố lớn; trong trường số nhị phân $F(2^m)$ thì *q* là 2^m, trong đó $m \geq 1$ là số nguyên dương.
- r là ước số nguyên tố lớn #E, với độ lớn ít nhất là 160 bit.
- $k = \#E/r$. Để giảm độ phức tạp trong việc kiểm tra tính hợp lệ của thông điệp nhận được trong cơ chế thiết lập khóa được quy định trong tiêu chuẩn này, yêu cầu thỏa mãn $k = 2^n p_1 p_2 \dots p_t$ trong đó n = {0, 1, 2}, với số nguyên tố $p_i > r$, $i = 1, 2, \dots, t$. Tùy chọn $k = 2^n$.

CHÚ THÍCH Thông tin chung về đường cong elliptic trên *F(q)* có thể tìm thấy trong ISO/IEC 15946–1.

C.2 Tham số trong cơ chế lấy lại khóa 1

Điều quan trọng trong cơ chế lấy lại khóa 1 quy định tại Điều 7.1, là cung cấp cơ chế tiên đoán Static Diffie–Hellman (*SDH*) cho kẻ tấn công đã bị khai thác trong tấn công *SDH* của [BG04]. Kiểu tấn công này cải thiện đáng kể khả năng của kẻ tấn công nhằm biết được khóa bí mật *x* của SDH, trước đó đã được chấp nhận truy cập vào thẻ cung cấp g^x cho giá trị g. Trong trường hợp "tối ưu", lực ngoại tuyến của kẻ tấn công là nhỏ nhất, lực này là giảm từ $r^{1/2}$ xuống $r^{1/3}$, trong đó *r* là số bậc của phần tử *g*.

Tuy nhiên, trong trường hợp "tối ưu" này sử dụng kiểu tấn công *SDHP* sẽ đòi hỏi tới $r^{1/3}$ thẻ truy vấn, điều này xem như là không thực tế và đặc biệt là không thực tế đối với phương pháp dùng mật khẩu. Vì vậy, kết quả mới này có nghĩa hay không phụ thuộc vào các tham số miền và cách mà sơ đồ này được sử dụng.

Có một số cách lựa chọn giá trị r để tránh kiểu tấn công SDH:

1. Giá trị *r* được chọn sao cho không tồn tại nghiệm của phương trình $r = uv + 1$ với các tính chất $1 \le u < a$ và $1 \le v < b$, trong đó *a* và *b* là các số nguyên nhỏ nhất sao cho lũy thừa *a/2* và dẫn chứng *b* của cơ chế là được coi không thể tính toán được.

2. Sử dụng nhóm lớn hơn, ví dụ kích thước của *r* là 240 bit thay vì chỉ lớn hơn 160 bit.

3. Đảm bảo rằng $(r - 1)$ không có ước số trong khoảng $[C, r^{1/2}]$. Tại điểm cuối thấp nhất, *C* là giá trị nhỏ "cắt rời". Tại điểm cuối cao nhất, $r^{1/2}$ là hầu như bị phá hủy, nhưng lại dễ nhớ. Khi giá trị $(r - 1)$ không có ước số trong phạm vi này, các cuộc tấn công mới không lợi thế hơn các phương pháp trước đó. Gợi ý là $(r - 1)/2$ là một số nguyên tố. Chú ý rằng điều này khác với trường hợp $k = 2$ trong cấu hình *DL*.

Tài liệu tham khảo

[1] [ANSI X9.62] American National Standards Institute. Public key cryptography for the financial services industry: the elliptic curve digital signature algorithm. ANSI X9.62:1998,January 1999. (*ANSI X9.62 của Viện Tiêu chuẩn quốc gia Mỹ. Mật mã hóa khóa công khai cho ngành dịch vụ tài chính: thuật toán ký số đường cong elliptic*).

[2] [BG04] D. Brown and R. Gallant, The Static Diffie–Hellman Problem. Cryptology ePrint Archive: Report 2004/306, 15 Nov 2004. (*Báo cáo về các cơ chế tiên đoán Static Diffie-Hellman*).

[3] [FK00] W. Ford and B. Kaliski. Server–assisted generation of a strong secret from a password. In the Proceedings of the IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp 176 – 180, IEEE, June 2000. (*Việc tạo bảo mật mạnh từ mật khẩu trợ giúp cho máy chủ. Báo cáo trong hội thảo quốc tế IEEE lần thứ 9*).

[4] [IEEEP1363] IEEE P1363. Standard for public key cryptography, 2000. (*Chuẩn về mật mã hóa khóa công khai*).

[5] [IEEEP1363a] IEEE 1363A–2004: Standard Specifications For Public Key Cryptography — Amendment 1: Additional Techniques. (*Các đặc tả cho mật mã hóa khóa công khai – Bổ sung 1: Các kỹ thuật bổ sung*).

[6] [IEEEP1363.2] IEEE P1363.2/D21:2005–07–17. Standard specifications for password–based public key cryptographic techniques. (*Các đặc tả cho các kỹ thuật mật mã hóa khóa công khai dựa trên mật khẩu*).

[7] [ISO/IEC 9796–3:2000] ISO/IEC 9796–3:2000, Information technology — Security techniques — Digital signature schemes giving message recovery — Part 3: Discrete logarithm based mechanisms. (*Công nghệ thông tin – Kỹ thuật an ninh – Lược đồ chữ ký số để phục hồi thông điệp – Phần 3: Cơ chế dựa trên logarit rời rạc*).

[8] [ISO/IEC 9798–3:1998] ISO/IEC 9798–3:1998, Information technology — Security techniques – Entity authentication — Part 3: Mechanisms using digital signature techniques. (*Công nghệ thông tin – Kỹ thuật an ninh – Xác thực thực thể - Phần 3: Cơ chế sử dụng kỹ thuật chữ ký số*).

[9] [ISO/IEC 10118–1:2000] ISO/IEC 10118–1:2000, Information technology — Security techniques — Hash–functions — Part 1: General. (*Công nghệ thông tin – Kỹ thuật an ninh – Hàm băm – Phần 1: Khái quát*).

[10] [TCVN 7817 – 3:2007] TCVN 7817 – 3:2007 Công nghệ thông tin – Kỹ thuật mật mã – Quản lý khóa, phần 3: Các cơ chế sử dụng kỹ thuật không đối xứng.

[11] [ISO/IEC 15946–1:2002] ISO/IEC 15946–1:2002, Information technology — Security techniques — Cryptographic techniques based on elliptic curves — Part 1: General. (*Công nghệ thông tin – Kỹ thuật an ninh – Kỹ thuật mật mã hóa dựa trên đường cong elliptic*).

[12] [ISO/IEC 15946–3:2002] ISO/IEC 15946–3:2002, Information technology — Security techniques — Cryptographic techniques based on elliptic curves — Part 3: Key establishment. (*Công nghệ thông tin – Kỹ thuật an ninh – Kỹ thuật mật mã hóa dựa trên đường cong elliptic – Phần 3: Thiết lập khóa*).

[13] [ISO/IEC 18031:2005] ISO/IEC 18031:2005, Information technology — Security techniques — Random bit generation. (*Công nghệ thông tin – Kỹ thuật an ninh – Tạo bit ngẫu nhiên*).

[14] [ISO/IEC 18032:2005] ISO/IEC 18032:2005, Information technology — Security techniques — Prime number generation. (*Công nghệ thông tin – Kỹ thuật an ninh – Tạo số nguyên tố*).

[15] [ISO/IEC 18033–1:2005] ISO/IEC 18033–1:2005, Information technology — Security techniques — Encryption algorithms — Part 1: General. (*Công nghệ thông tin – Kỹ thuật an ninh – Thuật toán mã hóa – Phần 1: Khái quát*).

[16] [ISO/IEC 18033–2] ISO/IEC 18033–2, Information technology — Security techniques — Encryption algorithms — Part 2: Asymmetric ciphers. (*Công nghệ thông tin – Kỹ thuật an ninh – Thuật toán mã hóa – Phần 3: Mật mã phi đối xứng*).

[17] [Jab96] D. Jablon. Strong password–only authenticated key exchange. Computer Communication Review, ACM SIGCOMM, 26(5):5–26, October 1996. (*Trao đổi khóa xác thực bằng mật khẩu mạnh*).

[18] [Ka86] B. Kaliski. A pseudo random bit generator based on elliptic logarithms. In Advances in Cryptology –CRYPTO '86, A. M. Odlyzko, Ed., vol. 263 of Lecture Notes in Computer Science, pp. 84–103, Springer–Verlag, 1987. (*Bộ tạo bit giả ngẫu nhiên dựa trên thuật toán logarit elliptic)*.

[19] [Kw00] T. Kwon. Ultimate solution to authentication via memorable password. Submission to the IEEE P1363 study group for future PKC standards, May 30, 2000. (*Giải pháp xác thực thông qua mật khẩu có thể nhớ*).

[20] [Kw03] T. Kwon. Addendum to summary of AMP. Submission to the IEEE P1363 study group for future PKC standards, November 19, 2003. (*Các phần bổ sung cho AMP*).

[21] [MvV96] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone. Handbook of Applied Cryptography. CRC Press, 1996. (*Hướng dẫn áp dụng mật mã hóa*).

[22] [TC05] Q. Tang and C. Mitchell. On the security of some password–based key agreement schemes. Cryptology ePrint Archive: Report 2005/156. (*Các vấn đề an ninh về lược đồ thỏa thuận khóa dựa trên mật khẩu*).

[23] [Wu02] T. Wu. SRP–6: improvements and refinements to the secure remote password protocol. Submission to IEEE P1363 Working Group, October 29, 2002. (*Tăng cường và tinh chỉnh cho các giao thức đảm bảo mật khẩu từ xa*).