QCVN 6 : 2016/BQP

QUY CHUẨN KỸ THUẬT QUỐC GIA VỀ QUẢN LÝ KHÓA SỬ DỤNG TRONG LĨNH VỰC NGÂN HÀNG

National technical regulation on key management used in banking

MỤC LỤC

Lời nói đầu

QCVN 6 : 2016/BQP do Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã - Ban Cơ yếu Chính phủ biên soạn, Ban Cơ yếu Chính phủ trình duyệt, Bộ Khoa học và Công nghệ thẩm định và được ban hành theo Thông tư số 161/2016/TT-BQP ngày 21 tháng 10 năm 2016 của Bộ trưởng Bộ Quốc phòng.

QUY CHUẨN KỸ THUẬT QUỐC GIA VỀ QUẢN LÝ KHÓA SỬ DỤNG TRONG LĨNH VỰC NGÂN HÀNG

National technical regulation on key management used in banking

1. QUY ĐỊNH CHUNG

1.1. Phạm vi điều chỉnh

Quy chuẩn kỹ thuật quốc gia này quy định các yêu cầu về quản lý khóa mật mã sử dụng kỹ thuật mật mã phi đối xứng để bảo mật dữ liệu trong lĩnh vực ngân hàng bao gồm: Sinh khóa bí mật dùng để liên lạc giữa hai thực thể bằng cơ chế thỏa thuận khóa và dẫn xuất khóa, sử dụng kỹ thuật mật mã phi đối xứng; sinh khóa bí mật cho một thực thể bởi thực thể khác bằng cơ chế truyền khóa sử dụng mật mã phi đối xứng; vận chuyển khóa công khai của một thực thể đến một thực thể khác bằng đường truyền có bảo vệ; quy định về quản lý sử dụng khóa mật mã an toàn.

Quy định về tạo khóa, đăng ký khóa, thu hồi, cài đặt, khôi phục và các vấn đề thuộc quản lý khóa khác không thuộc phạm vi điều chỉnh của quy chuẩn này.

1.2. Đối tượng áp dụng

Quy chuẩn này áp dụng đối với các doanh nghiệp kinh doanh sản phẩm, dịch vụ mật mã dân sự trong lĩnh vực ngân hàng; các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) sử dụng sản phẩm, dịch vụ mật mã dân sự.

1.3. Tài liệu viện dẫn

- TCVN 7817-3:2007 (ISO/IEC 11770-3:1999) Công nghệ thông tin - Kỹ thuật mật mã - Quản lý khóa - Phần 3: Các cơ chế sử dụng kỹ thuật phi đối xứng.

- ISO/IEC 11770-3:2015 Information technology - Security techniques - Key management - Part 3: Mechanisms using asymmetric techniques.

- Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, SP 800-56A Revision 2, National Institute of Standards and Technology, May 2013.

- *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, NIST SP 800-90A Rev. 1, National Institute of Standards and Technology, June 2015. *(Khuyến cáo cho bộ sinh số ngẫu nhiên sử dụng bộ sinh bit ngẫu nhiên tất định*, NIST SP 800-90A Rev. 1, Viện tiêu chuẩn và công nghệ quốc gia (Mỹ), tháng 6 năm 2015).

1.4. Giải thích từ ngữ

Trong Quy chuẩn này, các từ ngữ dưới đây được hiểu như sau:

1.4.1. Thông tin không thuộc phạm vi bí mật nhà nước

Là thông tin không thuộc nội dung tin “tuyệt mật”, “tối mật” và “mật” được quy định tại Pháp lệnh Bảo vệ bí mật nhà nước ngày 28 tháng 12 năm 2000.

1.4.2. Mật mã

Là những quy tắc, quy ước riêng dùng để thay đổi hình thức biểu hiện thông tin nhằm bảo đảm bí mật, xác thực, toàn vẹn của nội dung thông tin.

1.4.3. Mật mã dân sự

Là kỹ thuật mật mã và sản phẩm mật mã được sử dụng để bảo mật hoặc xác thực đối với thông tin không thuộc phạm vi bí mật nhà nước.

1.4.4. Sản phẩm mật mã dân sự

Là các tài liệu, trang thiết bị kỹ thuật và nghiệp vụ mật mã để bảo vệ thông tin không thuộc phạm vi bí mật nhà nước.

1.4.5. Kỹ thuật mật mã

Là phương pháp, phương tiện có ứng dụng mật mã để bảo vệ thông tin.

1.4.6. Mã hóa

Phép biến đổi (khả nghịch) dữ liệu bởi thuật toán mật mã để tạo ra bản mã, tức là giấu nội dung thông tin của dữ liệu.

1.4.7. Giải mã

Phép toán ngược với phép mã hóa tương ứng.

1.4.8. Mã phi đối xứng

Hệ thống dựa trên kỹ thuật mật mã phi đối xứng, trong đó phép biến đổi công khai được sử dụng để mã hóa, phép biến đổi bí mật được sử dụng để giải mã.

1.4.9. Kỹ thuật mật mã phi đối xứng

Kỹ thuật mật mã phi đối xứng sử dụng hai phép biến đổi liên quan đến nhau, phép biến đổi công khai (được xác định bởi khóa công khai) và phép biến đổi bí mật (được xác định bởi khóa riêng). Cả hai phép biến đổi này có tính chất là cho biết phép biến đổi công khai, về mặt tính toán không thể có khả năng xác định được phép biến đổi bí mật.

1.4.10. Thẻ khóa

Thông điệp quản lý khóa được gửi từ một thực thể tới một thực thể khác trong quá trình thực hiện một cơ chế quản lý khóa.

1.4.11. Vận chuyển khóa

Tiến trình truyền một khóa từ một thực thể đến một thực thể khác với bảo vệ thích hợp.

1.4.12. Xác thực thực thể lẫn nhau

Sự xác thực giữa hai thực thể đảm bảo về định danh của mỗi thực thể.

1.4.13. Xác thực khóa từ thực thể *A* đến thực thể *B*

Đảm bảo cho B rằng chỉ có *A* là thực thể sở hữu khóa đúng.

1.4.14. Xác thực khóa hai chiều

Đảm bảo xác thực khóa từ *A* đến *B* và từ *B* đến *A*.

1.4.15. Xác nhận khóa từ *A* đến *B*

Đảm bảo cho thực thể *B* là thực thể *A* sở hữu khóa đúng.

1.4.16. Xác nhận khóa hai chiều

Đảm bảo xác nhận khóa từ *A* đến *B* và từ *B* đến *A.*

1.4.17. Thiết lập khóa

Quá trình đảm bảo sự khả dụng một khóa bí mật dùng chung cho một hoặc nhiều thực thể. Thiết lập khóa bao gồm thỏa thuận khóa và vận chuyển khóa.

1.4.18. Thoả thuận khóa

Tiến trình kiến tạo một khóa bí mật dùng chung giữa hai thực thể theo cách mà không có bên nào có thể định trước giá trị cho khóa

1.5. Các ký hiệu

ID_A, ID_B	Định danh của các thực thể *A* và *B*
$Cert_x$	Chứng chỉ khóa công khai của thực thể X
F(h,g)	Hàm thỏa thuận khóa
HASH	Hàm băm
DH	Diffie-Hellman
MQV	Menezes-Qu-vanstone
K	Khóa bí mật cho hệ mật đối xứng
K_{AB}	Khóa bí mật chia sẻ giữa hai thực thể *A* và *B*
KT_{Ai}	Thông báo thỏa thuận khóa được gửi bởi thực thể *A* sau giai đoạn xử lý *i*
r	Số ngẫu nhiên được sinh trong quá trình thực hiện của một lược đồ.
H	Tập các phần tử có thể của *r*
S_x	Hàm tạo chữ ký sử dụng khóa riêng của thực thể *X*
V_x	Hàm kiểm tra chữ ký của thực thể *X*
||	Phép nối hai phần tử dữ liệu với nhau
x	Số nguyên nhỏ nhất lớn hơn hoặc bằng số thực *x*
len(x)	Độ dài số nguyên *x* tính bằng bit
nlen	Độ dài modulo *n* tính theo bit
(n, e)	Khóa công khai theo RSA
(n, d)	Khóa riêng theo RSA
L	Độ dài số nguyên tố *p* trong bài toán logarit rời rạc
N	Độ dài số nguyên tố *q* trong bài toán logarit rời rạc
q	Ước nguyên tố của *p* - 1

#(E)	Cấp hay lực lượng của đường cong elliptic *E*
m	Bậc của điểm sinh *G*
X(P)	Hoành độ *x* của điểm *P* trên đường cong elliptic *E*
h	Các đồng thừa số được tính theo công thức $h = \#E/m$
I	Thừa số phụ trong phép nhân đồng thừa số $I = h^{-1} mod\ m$
(P)	Phép biến đổi điểm *P* trên đường cong *E* thành số nguyên $\pi(P) = \left(X(P) mod 2^{\lfloor \rho/2 \rfloor}\right) + 2^{\lfloor \rho/2 \rfloor}, \rho = \lceil log_2 n \rceil$

2. QUY ĐỊNH KỸ THUẬT

2.1. Các yêu cầu đối với giao thức thỏa thuận và vận chuyển khóa

Điều này quy định các giao thức thiết lập khóa bí mật dùng để mã dữ liệu được trao đổi giữa hai thực thể *A* và *B*, vận chuyển khóa bí mật từ thực thể *A* sang thực thể *B*, vận chuyển khóa công khai của thực thể *A* sang thực thể *B*.

Để thực hiện giao thức, mỗi thực thể *X* đảm bảo các điều kiện sau:

- Sở hữu cặp khóa để ký và kiểm tra chữ ký (S_X, V_X) được cơ quan thẩm quyền cấp dưới dạng chứng thư số $Cert_X$.

- Sử dụng một cặp khóa công khai để mã hóa (E_X, D_X) được quy định tại Điều 2.7 của Quy chuẩn này.

- Sử dụng chung với thực thể thứ hai hàm thỏa thuận khóa *F* là một trong hai hàm được xác định tại Điều 2.2.2, một hàm dẫn xuất khóa *KDF* được quy định tại Điều 2.4 của Quy chuẩn này và một hàm kiểm tra mật mã $MAC_{K_{AB}}$ dưới dạng hàm băm được quy định tại Điều 2.2 của QCVN 5 : 2016/BQP Quy chuẩn kỹ thuật quốc gia về Chữ ký số sử dụng trong lĩnh vực ngân hàng.

- Mỗi thực thể được tiếp cận các khóa công khai của thực thể kia theo cơ chế vận chuyển khóa công khai tại Điều 2.6.

2.2. Giao thức thỏa thuận khóa sử dụng kỹ thuật mật mã phi đối xứng trên trường hữu hạn

Thỏa thuận khóa bí mật giữa hai thực thể *A* và *B* được thực hiện qua 5 bước:

Bước 1-4. Hai bên thỏa thuận bí mật chia sẻ K_{AB}.

Bước 5. Hai bên sử dụng hàm dẫn xuất khóa *KDF* được quy định tại Điều 2.4 để thiết lập khóa bí mật chung *K*.

Lược đồ thỏa thuận khóa bí mật chia sẻ K_{AB} được thể hiện trên Hình 1.

2.2.1. Giao thức

Bước 1. Kiến thiết thẻ khóa (A1): Thực thể A sinh một giá trị ngẫu nhiên và bí mật r_A thuộc *H,* tính $F(r_A,g)$, kiến thiết thẻ khóa KT_{A1} và gửi tới thực thể *B*:

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

Bước 2. Xử lý thẻ khóa và Kiến thiết khóa (B1): Thực thể *B* sinh một giá trị ngẫu nhiên và bí mật r_B thuộc *H*, tiếp đó tính $F(r_B, g)$, rồi tính khóa bí mật dùng chung:

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

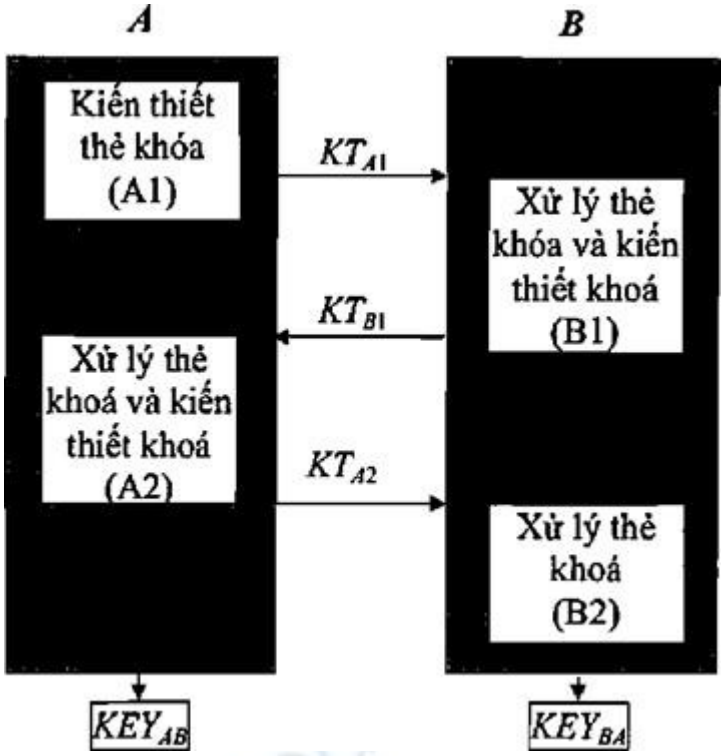
Tiếp theo, thực thể *B* tạo thẻ khóa KT_{B1} được ký như sau:

$$KT_{B1} = S_B(DB_1) \,||\, MAC_{K_{AB}}(DB_1) \,||\, Text3$$

trong đó

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

và gửi ngược trở lại cho thực thể A.



Hình 1: Giao thức thỏa thuận khóa bí mật chia sẻ K_{AB}

Bước 3. Xử lý thẻ khóa (A2): Thực thể *A* kiểm tra chữ ký của thực thể *B* trên thẻ khóa KT_{B1} bằng cách sử dụng khóa kiểm tra công khai của *B*, kiểm tra định danh phân biệt của *A* và giá trị $F(r_A,g)$ đã được gửi ở Bước 1 **(A1)**. Nếu quá trình kiểm tra thành công thì thực thể *A* sẽ tính khóa bí mật dùng chung là:

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

Thực thể *A* lại sử dụng khóa bí mật dùng chung K_{AB} để kiểm tra giá trị kiểm tra mật mã $MAC_{K_{AB}}(DB_1)$. Sau đó thực thể *A* tạo thẻ khóa KT_{A2} được ký như sau:

$$KT_{A2} = S_A(DB_2) \,||\, MAC_{K_{AB}}(DB_2) \,||\, Text5$$

trong đó

$$DB_2 = F(r_A,g) \,||\, F(r_B,g) \,||\, ID_B \,||\, Text4$$

và gửi thẻ này tới thực thể *B*.

Bước 4. Xử lý thẻ khóa (B2): Thực thể *B* kiểm tra chữ ký của thực thể *A* trên thẻ khóa KT_{A2} sử dụng khóa kiểm tra công khai của *A*, sau đó kiểm tra định danh phân biệt của *B* và kiểm tra các giá trị $F(r_A,g)$, và $F(r_B,g)$, xem có phù hợp với các giá trị được trao đổi ở các bước trước hay không. Nếu quá trình kiểm tra thành công thì thực thể *B* sẽ kiểm tra giá trị kiểm tra mật mã $MAC_{K_{AB}}(DB_2)$ bằng cách tính:

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

Bước 5. Thực thể *A* và thực thể *B* tính khóa bí mật chung *K*

Các thực thể *A* và *B* sử dụng hàm dẫn xuất khóa tại Điều 2.4 để tính khóa

$$K = KDF(K_{AB}, OtherInput)$$

Trong đó *OtherInput* chứa các định danh ID_A, ID_B của *A,B* và các thông tin khác do *A* và *B* thỏa thuận.

CHÚ THÍCH: Các trường *Text1, Text2, Text3, Text4, Text5* là những trường dữ liệu chứa chứng thư số của *A* và *B*, và có thể một số thông tin khác như tem thời gian, định danh phiên liên lạc, v.v..)

2.2.2. Hàm thỏa thuận khóa

2.2.2.1. Hàm thỏa thuận khóa *DH*

Cho trường hữu hạn nguyên tố F_p, ký hiệu tập $H = \{1, ...,p - 2\}$; *g* là phần tử sinh thuộc hàm thỏa thuận khóa Diffie-Hellman (*DH*) được xác định theo công thức:

$$F(h,g) = g^h,h \quad H$$

CHÚ THÍCH: Với hàm thỏa thuận *DH* thì thẻ khóa KT_{A1} có dạng :

$$KT_{A1} = g^{r_A} \| Text_A$$

2.2.2.2. Hàm thỏa thuận khóa MQV

- *(a,A); (b, B)* là cặp khóa phi đối xứng tĩnh của *A* và *B*, $A = g^a$, $B = g^b$

- *(x,X),(y,Y)* là cặp khóa phi đối xứng tức thời của *A* và *B*, $X = g^x, Y = g^y$, $d = 2^l + (X \bmod 2^l)$, $e = 2^l + (Y \bmod 2^l)$, $l = |p|/2$, *l* độ dài *p* tính bằng bit, $F(x, g) = (YB^e)^{x+da} = (XA^d)^{y+eb}$

2.3. Giao thức thỏa thuận khóa sử dụng mật mã trên đường cong elliptic

2.3.1. Giao thức DH

Bước 1. Kiến thiết thẻ khóa (A1)

Thực thể *A* chọn ngẫu nhiên và bí mật số r_A thuộc khoảng {2,…, *m* - 2}, tính r_AG, kiến thiết thẻ khóa $KT_{A1} = r_AG$ và gửi cho *B*.

Bước 2. Xử lí thẻ khóa và kiến thiết thẻ khóa (B1)

Thực thể *B* kiểm tra thẻ khóa KT_{A1} có phải là điểm nằm trên đường cong Elliptic hay không (kiểm tra theo tiêu chuẩn ISO/IEC 15946-1). Thực thể B chọn ngẫu nhiên và bí mật r_B thuộc khoảng {2, …, *m*-2}, tính r_BG, tính bí mật chia sẻ $K_{AB} = (r_Bl)(hKT_{A1})$, kiến thiết thẻ được ký hiệu KT_{B1}:

$$KT_{B1} = S_B(DB_1) \| MAC_{K_{AB}}(DB_1) \text{ với } DB_1 = r_BG \| KT_{A1} \| ID_A \| Text1$$

và gửi cho *A*.

Bước 3. Xử lí thẻ khóa (A2)

Thực thể *A* kiểm tra chữ ký của *B* trên thẻ khóa KT_{B1} sử dụng khóa kiểm tra công khai của *B*. Nếu sử dụng lược đồ chữ ký có khôi phục bản rõ thì việc kiểm tra bao gồm cả việc khôi phục khối dữ liệu DB_1 từ chữ ký và kiểm tra liệu định danh phân biệt của *A* và giá trị r_AG có chứa trong DB_1 không. Nếu sử dụng chữ ký có đính kèm bản rõ thì việc kiểm tra bao gồm cả việc thiết kế lại khối dữ liệu DB_1 sử dụng giá trị trong KT_{A1}, định danh phân biệt của *A*, giá trị nhận được r_BG và kiểm tra chữ ký trên khối dữ liệu này.

Tiếp đó thực thể *A* kiểm tra, liệu giá trị r_BG nhận được từ KT_{B1} có phải là điểm trên đường cong elliptic hay không (kiểm tra theo tiêu chuẩn ISO/IEC 15946-1). Nếu đúng thì *A* tính khóa chia sẻ $K_{AB} = (r_A.l)(h.r_BG)$.

Sử dụng K_{AB}, thực thể *A* kiểm tra $MAC_{K_{AB}}(DB_1)$. Tiếp đó *A* thiết kế thẻ khóa có ký hiệu $KT_{A2} = S_A(DB_2) \| MAC_{K_{AB}}(DB_2)$ ở đây $DB_2 = r_AG \| r_BG \| B \| Text2$ $(DB_2 = r_AG \| r_BG \| ID_B \| Text2)$ và gửi cho B.

Bước 4. Xử lí thẻ khóa (B2)

Thực thể *B* kiểm tra chữ ký của *A* trên thẻ khóa KT_{A2} bằng cách sử dụng khóa kiểm tra công khai của *A*. Nếu sử dụng sơ đồ chữ ký có khôi phục bản rõ thì điều này bao gồm cả việc khôi phục khối dữ liệu DB_2 từ chữ ký và kiểm tra định danh phân biệt của *B*, các giá trị r_AG và r_BG có chứa trong khối này không. Nếu sử dụng chữ ký có đính kèm bản rõ thì việc kiểm tra bao gồm cả việc thiết kế lại khối dữ liệu DB_2 sử dụng các giá trị trong KT_{A1} và KT_{B1}, định danh phân biệt của *B* và kiểm tra chữ ký trên khối dữ liệu này.

Nếu việc kiểm tra thành công thì thực thể B kiểm tra $MAC_{K_{AB}}(DB_2)$ sử dụng khóa chia sẻ $K_{AB} = (r_Bl)(hKT_{A1})$.

Bước 5. Thiết lập khóa bí mật K

Thực thể *A* và *B* sử dụng hàm dẫn xuất khóa *KDF* tại Điều 2.4 để thiết lập khóa bí mật.

$$K = KDF(K_{AB}, OtherInput).$$

2.3.2. Thỏa thuận khóa *MQV*

Bước 1. Thiết kế thẻ khóa (A1)

Thực thể *A* chọn ngẫu nhiên và bí mật số r_A thuộc khoảng {2,…,*m* - 2} tính r_AG và kiến thiết thẻ khóa $KT_{A1} = r_AG$ và gửi cho thực thể *B*.

Bước 2. Kiến thiết thẻ khóa (B1)

Thực thể *B* kiểm tra thẻ KT_{A1} có phải là điểm trên đường cong elliptic (kiểm tra theo tiêu chuẩn ISO/IEC 15946-1:2016). Thực thể *B* chọn ngẫu nhiên và bí mật số r_B thuộc khoảng {2,...,m - 2}, tính r_BG, thiết kế thẻ khóa $KT_{B1} = r_BG$, tiếp đó tính khóa bí mật chia sẻ K_{AB}:

$$K_{AB} = ((r_B + n(KT_{B1})d_B).l)(h.(KT_{A1} + (KT_{A1})P_A))$$

Tiếp đó *B* tính $K = KDF(K_{AB})$ và gửi $f_{K_{AB}}(2,KT_{A1},KT_{B1})$ gửi $MAC_K(2,KT_{A1},KT_{B1})$ cho A cùng với thẻ khóa KT_{B1}.

Bước 3. Kiến thiết thẻ khóa (A2)

Thực thể *A* tính khóa bí mật chia sẻ :

$$K_{AB} = ((r_A + (KT_{A1})d_A).l)(h.(KT_{B1} + (KT_{B1})P_B))$$

và kiểm tra $MAC_K(2,KT_{A1},KT_{B1})$

Tiếp đó A tính $MAC_K(3,KT_{A1},KT_{B1})$ và gửi cho *B*.

Bước 4. Kiểm tra (B2)

Thực thể *B* tính $MAC_K(3,KT_{A1},KT_{B1})$ và kiểm tra thực thể *A*.

2.4. Hàm dẫn xuất khóa KDF

2.4.1 Hàm dẫn khóa 1

Dạng thức của KDF

KDF(*Z, OtherInput*) trong đó *OtherInput* là *keydatalen* và *OtherInfo* (các đại lượng này được giải thích về sau).

Các giá trị cố định:

hashlen: số nguyên chỉ độ dài đầu ra (theo bit) của hàm băm được sử dụng để dẫn xuất ra các khối của dữ liệu khóa bí mật.

max_hash_inputlen: số nguyên có giá trị là độ dài lớn nhất (theo bit) của (các) chuỗi bit đầu vào của hàm băm.

Các hàm bổ trợ:

H: là hàm băm được chấp thuận là hàm băm được quy định tại QCVN 5 : 2016/BQP Quy chuẩn kỹ thuật quốc gia về Chữ ký số sử dụng trong lĩnh vực ngân hàng.

Đầu vào:

Z: Chuỗi byte bí mật chia sẻ trước.

keydatalen: số nguyên chỉ độ dài (theo bit) của dữ liệu khóa bí mật được sinh ra; *keydatalen* cần nhỏ hơn hoặc bằng *Hashlen* x $(2^{32} - 1)$

OtherInfo: Chuỗi bit sau:

$$AlgorithmID||PartyAInfo||PartyBInfo||SuppPubInfo||SuppPrivInfo\}$$

trong đó các trường con được định nghĩa như sau:

- *AlgorithmID*: Chuỗi bit chỉ ra cách thức phân tách dữ liệu khóa đã được dẫn xuất ra và dữ liệu khóa được dẫn xuất sẽ được sử dụng cho những thuật toán nào. Ví dụ, *AlgorithID* có thể chỉ ra rằng các bit 1-80 được dùng là 80-bit khóa cho *HMAC* và các bit 81-208 được dùng là 128-bit khóa cho AES.

- *PartyAInfo*: Một chuỗi bit chứa các thông tin công khai được yêu cầu bởi ứng dụng sử dụng hàm KDF được đóng góp bởi bên *A* trong quá trình dẫn xuất khóa, ở mức tối thiểu, *PartyAInfo* chứa ID_A là định danh bên *A*. Xem chú ý phần dưới.

- *PartyBInfo*: Một chuỗi bit chứa các thông tin công khai được yêu cầu bởi ứng dụng sử dụng hàm KDF được đóng góp bởi bên *B* trong quá trình dẫn xuất khóa, ở mức tối thiểu, *PartyVInfo* chứa ID_B là định danh bên *B*. Xem chú ý phần dưới.

- (Tùy chọn) *SuppPubInfo*: Một chuỗi bit chứa các thông tin công khai bổ sung cả hai bên cùng biết (mutual-known)

- (Tùy chọn) *SuppPrivInfo*: Một chuỗi bit chứa thông tin bí mật bổ sung cả hai bên cùng biết (mutual-known)

(Ví dụ, một khóa bí mật đối xứng chia sẻ trước được truyền thông qua một kênh riêng biệt)

Thuật toán:

1. Tính *reps* = $\lceil keydatalen/hashlen \rceil$.

2. Nếu (*reps* > $2^{32} - 1$) thì ABORT: chỉ thị lỗi và dừng.

3. Khởi tạo bộ đếm chuỗi bit 32-bit big-endian *counter* bằng 00000001_{16}.

4. Nếu *counter*||Z||*OtherInfo* có độ dài lớn hơn *max_hash_inputlen* thì chỉ thị lỗi và dừng lại.

5. Vòng lặp với *i* = 1 đến *reps*, thực hiện :