GIỚI THIỆU TIÊU CHUẨN AN TOÀN THÔNG TIN – MÃ HÓA CÓ SỬ DỤNG XÁC THỰC

Khi dữ liệu được gửi từ nơi này đến nơi khác thì cần phải bảo vệ dữ liệu trong quá trình đang được vận chuyển. Tương tự như vậy, khi dữ liệu được lưu trữ trong một môi trường mà các bên không được phép có thể truy cập thì cần thiết phải có các biện pháp bảo vệ dữ liệu đó. Bài báo sẽ giới thiệu tóm tắt nội dung tiêu chuẩn ISO/IEC 19772:2020 về an toàn thông tin – mã hóa có sử dụng xác thực. Xác định các cách thức xử lý một chuỗi dữ liệu theo các mục tiêu an toàn bao gồm 5 cơ chế mã hóa có sử dụng xác thực.

TỔNG QUAN VỀ MÃ HÓA CÓ SỬ DỤNG XÁC THỰC

Tính an toàn của dữ liệu cần được bảo vệ, ví dụ: để chống lại việc nghe trộm, thì một giải pháp là sử dụng mã hóa, như được quy định trong TCVN 11367 (ISO/IEC 18033) và ISO/IEC 10116. Ngoài ra, nếu cần bảo vệ dữ liệu chống lại sự sửa đổi, tức là bảo vệ tính toàn vẹn, thì mã xác thực thông điệp (MAC) như được chỉ định trong TCVN 11495 (ISO/IEC 9797), hoặc chữ ký số như được chỉ định trong TCVN 11495 (ISO/IEC 9797) và TCVN 12214 (ISO/IEC 14888) có thể được sử dụng. Nếu cả tính an toàn và tính toàn vẹn đều được yêu cầu, thì một khả năng là sử dụng cả mã hóa và MAC hoặc chữ ký. Mặc dù các hoạt động này có thể được kết hợp theo nhiều cách, nhưng không phải tất cả sự kết hợp của các cơ chế như vậy đều cung cấp các đảm bảo về độ an toàn giống nhau. Do đó, cần xác định chi tiết chính xác cách kết hợp các cơ chế toàn vẹn và an toàn để cung cấp mức độ an toàn tối ưu. Hơn nữa, trong một số trường hợp, có thể đạt được hiệu quả đáng kể bằng cách xác định một phương pháp xử lý dữ liệu duy nhất với mục tiêu cung cấp cả tính an toàn và tính toàn vẹn.

Trong tiêu chuẩn này, các cơ chế mã hóa được xác thực được xác định. Đây là những phương pháp xử lý dữ liệu để bảo vệ cả tính toàn vẹn và tính an toàn. Chúng thường liên quan đến sự kết hợp cụ thể của tính toán MAC và mã hóa dữ liệu hoặc sử dụng thuật toán mã hóa theo cách đặc biệt để cung cấp cả tính toàn vẹn và bảo vệ bí mật.

Các phương pháp được chỉ định trong tiêu chuẩn này đã được thiết kế để tối đa hóa mức độ an toàn và cung cấp khả năng xử lý dữ liệu hiệu quả. Một số kỹ thuật được định nghĩa ở đây có "chứng minh an toàn" toán học, tức là các lập luận chặt chẽ hỗ trợ tính hợp lý của chúng.

Tiêu chuẩn này quy định cụ thể 05 cơ chế mã hóa có xác thực, nghĩa là xác định các cách thức xử lý một chuỗi dữ liệu theo các mục tiêu an toàn:

– Tính bí mật dữ liệu, tức là bảo vệ chống lại truy cập dữ liệu trái phép.

– Tính toàn vẹn dữ liệu, tức là bảo vệ cho phép bên nhận xác minh dữ liệu nhận được không bị sửa đổi.

– Xác thực nguồn gốc dữ liệu, tức là bảo vệ cho phép bên nhận xác minh định danh bên gửi dữ liệu.

Tất cả 05 cơ chế được quy định trong tiêu chuẩn này dựa trên thuật toán mã khối, yêu cầu bên gửi và bên nhận dữ liệu được bảo vệ phải chia sẻ khóa bí mật cho mã khối này.

Quản lý khóa nằm ngoài phạm vi tiêu chuẩn này; các kỹ thuật quản lý khóa được quy định trong TCVN 7817 (ISO/IEC 11770).

Bốn trong số các cơ chế trong tiêu chuẩn này, cụ thể là cơ chế 3, 4, 5 (chỉ dành cho biến thể AAD) và 6, cho phép dữ liệu đã xác thực mã không cần phải mã hóa. Nghĩa là, các cơ chế này cho phép một chuỗi dữ liệu đã được bảo vệ được chia thành hai phần: D là chuỗi dữ liệu được mã hóa và được bảo vệ tính toàn vẹn; A (dữ liệu xác thực bổ sung) dùng để bảo vệ tính toàn vẹn nhưng không được mã hóa. Trong tất cả các trường hợp, chuỗi A có thể rỗng.

YÊU CẦU CỦA CƠ CHẾ MÃ HÓA CÓ XÁC THỰC

Các cơ chế mã hóa có xác thực được mô tả trong tiêu chuẩn này, bên gửi và bên nhận dữ liệu có áp dụng cơ chế mã hóa có xác thực phải thỏa mãn các yêu cầu sau:

– Thoả thuận sử dụng một cơ chế cụ thể từ các cơ chế được trình bày trong tiêu chuẩn này.

– Thỏa thuận sử dụng mã khối được sử dụng với cơ chế (sử dụng một trong các mã khối đã được chuẩn hóa trong TCVN 11367-3:2016 (ISO/IEC 18033-3:2010).

– Chia sẻ một khóa bí mật K: trong tất cả các cơ chế mã hóa có xác thực, ngoại trừ cơ chế 5, đây sẽ là khóa cho mã khối đã chọn và trong cơ chế 5, nó sẽ là khóa được sử dụng làm đầu vào cho hàm dẫn xuất khóa.

– Ngoài ra, mỗi cơ chế có các yêu cầu cụ thể được liệt kê ngay từ khi mô tả cơ chế.

CƠ CHẾ SỐ 1

Phần này trình bày về cơ chế mã hóa có xác thực thường được biết đến với tên gọi là bọc khóa (Key Wrap - KW).

Quy trình mã hóa

Bên gửi thực hiện theo các bước sau để bảo vệ chuỗi dữ liệu D:

– Chia D thành một chuỗi m khối 64-bit $D_1, D_2, \dots, D_m$, do đó D_1 chứa 64-bit đầu tiên của D, D_2 chứa 64-bit tiếp theo của D và tiếp tục cho đến hết D.

– Đặt Y là khối 64-bit biểu diễn theo hệ hexa A6A6A6A6A6A6A6A6 hoặc theo hệ nhị phân (10100110 10100110 … 10100110).

– Cho $i = 1,2, \ldots, m$:

• Đặt $R_i = D_i$.

– Cho $i = 1,2, \ldots, 6m$ thực hiện 04 bước sau:

○ Đặt $Z = e_K(Y \| R_1)$;

○ Đặt $Y = Z|_{64} \oplus \#_{64}(i)$;

○ Với $j = 1,2, \ldots, m$-1:

○ Đặt $R_j = R_{j+1}$;

○ Đặt $R_m = Z|^{64}$.

– Đặt $C_0 = Y$.

– Với $i = 1,2, \ldots, m$:

Đặt $C_i = R_i$.

Đầu ra của quá trình trên hay bản mã hóa có xác thực của D là chuỗi bit:

$$C = C_0 \| C_1 \| \ldots \| C_m$$

Hay C là một chuỗi $(64(m + 1))$-bit, nghĩa là C chứa nhiều hơn D 64-bit.

Quy trình giải mã

Bên nhận thực hiện các bước sau để giải mã và xác nhận một chuỗi C đã được mã hóa có xác thực.

– Nếu $len(C)$ không là một bội số của 64 hoặc nhỏ hơn 192, dừng và thông báo "Không hợp lệ".

– Chia C thành chuỗi $(m+1)$ khối 64 bit $C_0, C_1, \ldots, C_m$, do đó C_0chứa 64 bit đầu của C , C_1 chứa 64 bit tiếp theo và cứ thế tiếp tục.

– Đặt $Y = C_0$.

– Cho $i = 1,2, \ldots, m$:

• Đặt $R_i = C_i$.

– Cho $i = 6m, 6m$-1, giảm đi 1, thực hiện 04 bước sau:

○ Đặt $Z = d_K([Y \oplus \#_{64}(i)]||R_m)$;

○ Đặt $Y = Z|_{64}$;

○ Cho $j = m, m$-1, …, 2:

○ Đặt $R_j = R_{j+1}$;

○ Đặt $R_1 = Z|^{64}$.

– Nếu $Y = (10100110\ 10100110\ \ldots\ 10100110)$, thì đầu ra là $D = R_1 \| R_2 \| \ldots \| R_m$. Ngược lại thông báo "Không hợp lệ".

CƠ CHẾ SỐ 2

Phần này trình bày về quy trình mã hóa có xác thực thường được biết đến với tên gọi là CCM (Counter with CBC-MAC).

Quy trình mã hóa

Bên khởi tạo sẽ thực hiện các bước sau để bảo vệ chuỗi dữ liệu D. Đặt $L = \frac{len(D)}{8}$, tức là L là số octets trong D.

Quy trình giải mã

Bên nhận sẽ thực hiện các bước sau để giải mã và xác minh mã hóa - xác thực chuỗi C.

– Nếu C không chứa toàn bộ số octet thì tạm dừng và thông báo "Không hợp lệ".

– Nếu độ dài của C nhỏ hơn (t + 8)-bit, thì tạm dừng và thông báo "Không hợp lệ".

– Gọi m và r là các số nguyên duy nhất sao cho C chứa tổng cộng (128(m-1) + 8r + t)-bit, trong đó $0 < r \leq 16$. Chia C thành một dãy các khối: $C_1, C_2, \dots, C_m, U$ như sau. Đặt C_1 chứa 128-bit đầu tiên của C, C_2 chứa 128-bit tiếp theo của C, …, cho đến khi C_m chứa $8r$-bit tiếp theo của C. Cuối cùng, đặt U là t-bit cuối cùng của C.

– Đặt cờ octet $F = \left(0^5 \parallel \#_3(w-1)\right)$ và đặt $Y = (F \parallel S \parallel 0^{8w})$.

– Đặt $T = U \oplus [e_K(Y)]|_t$.

– Với $i = 1, 2, \dots, m-1$, thực hiện hai bước sau:

1) Đặt $Y = \left(F \parallel S \parallel \#_{8w}(i)\right)$;

2) Đặt $D_i = C_i \oplus e_K(Y)$.

– Đặt $Y = \left(F \parallel S \parallel \#_{8w}(m)\right)$, và đặt $D_m = C_m \oplus [e_K(Y)]|_{8r}$.

– Đặt $D = D_1 \parallel D_2 \parallel \cdots \parallel D_m$, và đặt $L = 16m - 16 + r$.

– Đệm bên phải D_m với 128-8r số "0", tức là đặt $D_m = D_m \,||0^{128-8r}$.

– Nếu $len(A) = 0$, thì đặt cờ octets $F = 0^2 \parallel \#_3\left(\frac{t-16}{16}\right) \parallel \#_3(w-1)$.

– Nếu $len(A) > 0$, thì đặt cờ octets $F = 0 \parallel 1 \parallel \#_3\left(\frac{t-16}{16}\right) \parallel \#_3(w-1)$.

– Đặt $X = e_K\left(F \parallel S \parallel \#_{8w}(L)\right)$.

– Nếu $len(A) > 0$, sau đó thực hiện 06

– Cho $i = 1, 2, \dots, m$:

Đặt $X = e_K(X \oplus D_i)$.

– Đặt $T' = X|_t$.

– Nếu $T = T'$, thì xuất D như đã tính ở bước h) và A. Ngược lại, thông báo "Không hợp lệ".

CƠ CHẾ SỐ 3

Phần này trình bày về cơ chế mã hóa có sử dụng xác thực thường được biết đến với tên gọi EAX.

Quy trình mã hóa

Bên gửi thực hiện các bước sau để bảo vệ chuỗi dữ liệu D:

– Lựa chọn biến khởi tạo S chứa n-bit. Biến S là riêng biệt đối với mỗi bản tin được bảo vệ, phải sẵn sàng tại bên nhận bản tin. Tuy nhiên, giá trị này không nhất thiết phải bí mật hoặc không thể đoán trước.

– Đặt $E_0 = M_K(0^n \| S)$.

– Đặt $E_1 = M_K(0^{n-1} \| 1 \| A)$.

– Đặt $W = E_0$.

– Chia D thành một chuỗi các khối: $D_1, D_2, \ldots, D_m$, như sau: đặt D_1 chứa n-bit đầu tiên của D, D_2 chứa n bit kế tiếp và cứ thế tiếp tục đến khi D_m chứa r-bit sau cùng, trong đó $0 < r \leq n$; Do đó $len(D) = (m\text{-}1)n + r$.

– Với $i = 1, 2, \ldots, m\text{-}1$, thực hiện 02 bước sau:

○ Đặt $C_i = D_i \oplus e_k(W)$.

○ Đặt $W = \#_n(\#^{-1}(W) + 1 mod 2^n)$.

– Đặt $C_m = D_m \oplus [e_K(W)]|_r$.

– Đặt $E_2 = M_k\ (0^{n-2} \| 1 \| 0 \| C_1 \| C_2 \| \ldots\ldots \| C_m)$.

– Đặt $= [E_0 \oplus E_1 \oplus E_2]|_t$.

Đầu ra của quá trình trên, tức là bản mã hóa có sử dụng xác thực của D là chuỗi bit:

$$C = C_1 \| C_2 \| \ldots \| C_{m-1} \| C_m \| T$$

Hay C là một chuỗi $((m-1)n + r + t)$-bit, nghĩa là chứa nhiều hơn t-bit so với chuỗi dữ liệu gốc D (mặc dù nó phải cần mang n-bit biến khởi tạo S và dữ liệu A đã được xác thực bổ sung có độ dài thay đổi đến bên nhận).

Quy trình giải mã

Bên nhận thực hiện các bước sau để giải mã và xác nhận một chuỗi C đã được mã hóa có xác thực.

– Nếu độ dài của C nhỏ hơn t, dừng và thông báo "Không hợp lệ".

– Đặt m và r là các số nguyên duy nhất xác định sao cho C chứa tổng $((m-1)n + r + t)$-bit, trong đó $0 < r \leq n$. Chia C thành một chuỗi các khối: $C_1, C_2, \ldots, C_m, T$. Đặt C_1 chứa n-bit đầu tiên của C, C_2 chứa n-bit kế tiếp và cứ như thế cho đến khi C_m chứa r-bit kế tiếp của C. Sau cùng, đặt T là t-bit cuối cùng của C.

– Đặt $E_0 = M_K(0^n \| S)$.

– Đặt $E_1 = M_K(0^{n-1} \| 1 \| A)$.

– Đặt $E_2 = M_K(0^{n-2} \| 1 \| 0 \| C_1 \| C_2 \| \ldots \| C_m)$.

– Đặt $T' = [E_0 \oplus E_1 \oplus E_2]|_t$.

– Nếu $T \neq T'$, thì dừng và thông báo "Không hợp lệ".

– Đặt $W = E_0$.

– Với $i = 1, 2, \ldots, m-1$, thực hiện 02 bước sau:

○ Đặt $D_i = C_i \oplus e_K(W)$;

○ Đặt $W = \#_n(\#^{-1}(W) + 1\ mod 2^n)$.

– Đặt $D_m = C_m \oplus [e_K(W)]|_r$.

– Đầu ra là D và A.

CƠ CHẾ SỐ 4

Phần này trình bày về cơ chế mã hóa có xác thực được tạo nên từ sự kết hợp giữa một cơ chế mã hóa và một lược đồ MAC được xác định (Encrypt-then-MAC). Lược đồ này yêu cầu mã hóa dữ liệu được bảo vệ trước, sau đó tính toán MAC trên kết quả dữ liệu đã được mã hóa.

Quy trình mã hóa

Bên gửi phải thực hiện các bước sau để bảo vệ chuỗi dữ liệu D và nếu sử dụng cơ chế biến thể AAD, để đảm bảo tính toàn vẹn của chuỗi dữ liệu A đã được xác thực bổ sung.

– Chọn biến khởi tạo S thích hợp để sử dụng với chế độ hoạt động của mã khối của các hoạt động sẽ chọn. Biến S là riêng biệt đối với mỗi thông điệp được bảo vệ theo khóa đã cho và phải sẵn sàng tại bên nhận thông điệp. Các yêu cầu có thể có đối với S được mô tả chi tiết trong các phần tương ứng của tiêu chuẩn TCVN 12213:2018 (ISO/IEC 10116:2017).

– Đặt $C' = \varepsilon_{K_1,S}\ \varepsilon(D)$.

o Nếu không sử dụng biến thể AAD:

– Đặt $T = f_{K_2}(S \parallel C')$.

– Nếu $len(A)$ không phải là bội của 8 hoặc $\geq\ 2^{67}$, sau đó tạm dừng và thông báo "Không hợp lệ".

Đặt $T = f_{K_2}(\left(\frac{len(A)}{8}\right) \parallel A \parallel S \parallel\ C')$.

Đầu ra của quá trình trên, tức là biến thể của D được mã hóa - xác thực, phải là chuỗi bit:

$C\ =\ C' \parallel\ T$, cùng với biến khởi tạo SS.

Quy trình giải mã

Bên nhận phải thực hiện các bước sau để giải mã và xác minh một chuỗi C được mã hóa - xác thực, với biến khởi tạo S kèm theo và nếu biến AAD sử dụng cơ chế để xác minh toàn vẹn của dữ liệu được xác thực bổ sung A.

– Nếu độ dài của C nhỏ hơn t thì dừng, thông báo "Không hợp lệ".

– Đặt T là t-bit ngoài cùng bên phải của C và đặt C' bằng C với t-bit ngoài cùng bên phải bị loại bỏ, tức là $C\ =\ C' \mid\mid t$..

o Nếu không sử dụng biến AAD:

– Đặt $T' =\ f_{K_2}(S \mid\mid C')$.

o Nếu sử dụng biến AAD:

– Nếu $len(A)$ không phải là bội của 8 hoặc $\geq\ 2^{67}$ thì tạm dừng và thông báo "Không hợp lệ".

Đặt $T' =\ f_{K_2}(\#_{64}\left(\frac{len(A)}{8}\right)\ \parallel\ A\ \parallel\ S\ \parallel\ C')$.

– Nếu $T\ \neq\ T'$ thì tạm dừng và và thông báo "Không hợp lệ".

– Đặt $D = \delta_{K_1,S}(C')$.
– Đầu ra là D.

CƠ CHẾ SỐ 5

Phần này trình bày về cơ chế mã hóa có xác thực thường được biết đến với tên gọi GCM (Galois/Counter Mode).

Quy trình mã hóa

Bên gửi thực hiện các bước sau để bảo vệ chuỗi dữ liệu D và đảm bảo tính toàn vẹn của chuỗi dữ liệu đã được xác thực bổ sung A.

– Chọn biến khởi tạo S có độ dài tùy ý. Giá trị biến S phải là riêng biệt đối với mỗi thông điệp được bảo vệ và phải sẵn sàng tại bên nhận thông điệp. Tuy nhiên, giá trị này không nhất thiết phải bí mật hoặc không thể đoán trước.
– Giá trị S có thể được tạo ra bằng cách sử dụng một bộ đếm được duy trì bởi bên gửi và gửi đi trong một bản gốc đính kèm với thông điệp đã được bảo vệ.
– Phân chia D thành một chuỗi các khối có kích thước 128-bit: $D_1, D_2, \ldots, D_m$. Đặt D_1 chứa 128-bit đầu tiên của D, D_2 chứa 128-bit kế tiếp và cứ thế tiếp tục cho đến khi D_m chứa r-bit sau cùng của D, trong đó $0 < r \leq 128$. Do đó, D chứa tổng số $((m-1)n + r)$-bit.
– Đặt $H = e_K(0^{128})$.
– Nếu $len(S) = 96$ thì đặt $Y_0 = S \parallel 0^{31} \parallel 1$. Cách khác, đặt $Y_0 = G(H, \{\}, S)$.
– Cho $i = 1, 2, \ldots, m-1$ thực hiện 02 bước sau:
○ Đặt $Y_i = inc(Y_{i-1})$;
○ Đặt $C_i = D_i \oplus e_K(Y_i)$.
– Đặt $Y_m = inc(Y_{m-1})$.
– Đặt $C_m = D_m \oplus (e_K(Y_m))|_r$.

Đầu ra của quá trình trên, biến thể mã hóa - xác thực của D phải là chuỗi bit:

$$C = C_1 \parallel C_2 \parallel \ldots \parallel C_m \parallel T$$

Hay C là một chuỗi $(m-1)n + r + t$, nghĩa là chuỗi C chứa nhiều hơn so với chuỗi dữ liệu gốc D t-bit (mặc dù nó phải cần mang thêm n-bit biến khởi tạo S và dữ liệu A đã được xác thực bổ sung có độ dài thay đổi đến bên nhận).

Quy trình giải mã

Bên nhận thực hiện các bước sau để giải mã và xác nhận một chuỗi C đã được mã hóa có sử dụng xác thực và xác nhận dữ liệu A đã được xác thực bổ sung.

– Nếu độ dài của C nhỏ hơn t thì dừng và thông báo "Không hợp lệ".
– Đặt m và r là các số nguyên duy nhất xác định sao cho C chứa một tổng $\big((m-1)n + r + t\big)$-bit, trong đó $0 < r \leq n$. Chia C thành một chuỗi các khối: $C_1, C_2, \ldots, C_m, T$. Đặt C_1 chứa n-bit đầu tiên của C, C_2 chứa n-bit kế tiếp và cứ như thế cho đến khi C_m chứa r-bit tiếp theo của C. Cuối cùng, đặt T là t-bit cuối cùng của C.

– Đặt $H = e_K(0^{128})$.
– Nếu $len(S) = 96$ thì đặt $Y_0 = S \parallel 0^{31} \parallel 1$. Cách khác, đặt $Y_0 = G(H, \{\}, S)$.
– Nếu $T' = (G(H, A, C_1 \parallel C_2 \parallel \ldots \parallel C_m) \oplus e_K(Y_0))|_t$.
– Nếu $T \neq T'$ thì dừng và thông báo "Không hợp lệ".
– Với $i = 1, 2, \ldots, m-1$, hãy thực hiện 02 bước sau:
○ Đặt $Y_i = inc(Y_{i-1})$;
○ Đặt $D_i = C_i \oplus e_K(Y_i)$.
– Đặt $Y_m = inc\,(Y_{m-1})$.
– Đặt $D_m = C_m \oplus (e_K(Y_m))|_r$.
– Đầu ra D và dữ liệu xác thực bổ sung A.

KẾT LUẬN

Tiêu chuẩn ISO/IEC 19772:2020 đang được Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã biên soạn, được Ban Cơ yếu Chính phủ đề nghị Tổng cục Tiêu chuẩn Đo lường Chất lượng thẩm định, Bộ Khoa học và Công nghệ công bố. Bài viết sẽ giới thiệu tổng quan nội dung của ISO/IEC 19772:2020, quy định chi tiết 5 cơ chế mã hóa có sử dụng xác thực, nghĩa là xác định các cách thức xử lý một chuỗi dữ liệu theo các mục tiêu an toàn: Tính bí mật dữ liệu, tức là bảo vệ chống lại truy cập dữ liệu trái phép; Tính toàn vẹn dữ liệu, tức là bảo vệ cho phép bên nhận xác minh dữ liệu nhận được không bị sửa đổi; Tính xác thực nguồn gốc dữ liệu, tức là bảo vệ cho phép bên nhận xác minh định danh bên gửi dữ liệu.