(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)**
CỤC SỞ HỮU TRÍ TUỆ

(11) **1-0042584**

(51)[2020.01] **G06F 21/00**

(13) **B**

(21) 1-2021-00705

(22) 08/02/2021

(45) 27/01/2025 442

(43) 25/08/2022 413

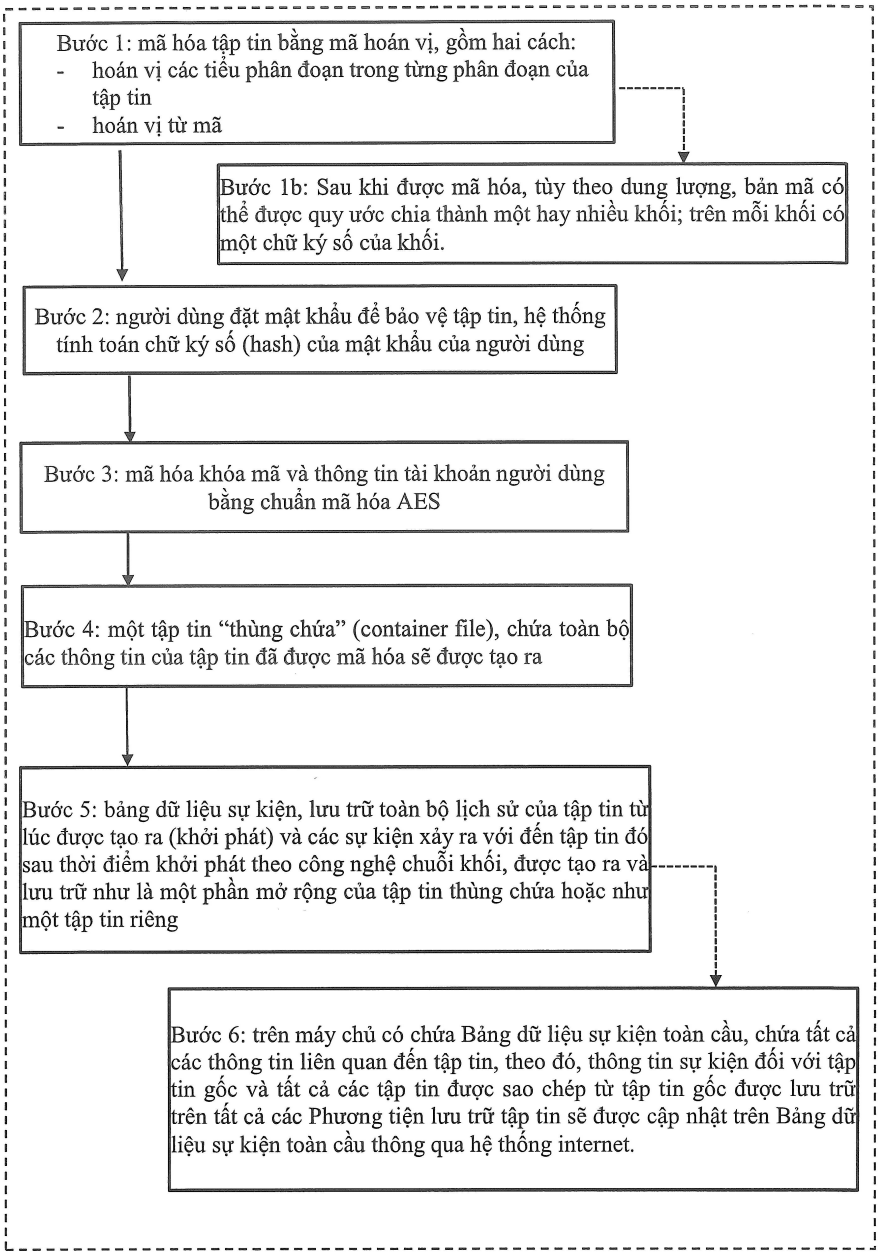
(76) Nguyễn Khương Tuấn (VN)
Số 72 Trần Cung, tổ 21, phường Nghĩa Tân, quận Cầu Giấy, thành phố Hà Nội, Việt Nam

(54) PHƯƠNG PHÁP BẢO VỆ NỘI DUNG TẬP TIN CÓ ENTROPY THÔNG TIN CAO BẰNG CÁCH SỬ DỤNG KẾT HỢP MÃ HOÁN VỊ, CHUẨN MÃ HÓA AES VÀ CÔNG NGHỆ CHUỖI KHỐI VÀ HỆ THỐNG THỰC HIỆN PHƯƠNG PHÁP NÀY

(21) 1-2021-00705

(57) Sáng chế đề xuất phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao bằng cách sử dụng kết hợp mã hoán vị, chuẩn mã hóa AES và công nghệ chuỗi khối, trong đó mã hoán vị có thể được thực hiện bằng cách cách hoán vị từng tiểu phân đoạn trong phân đoạn hoặc hoán vị từng từ mã trong không gian từ mã. Chuẩn mã hóa đối xứng gồm AES, Twofish, Serpent, Blowfish, CAST5, RC4, Tam phần DES (Triple DES), và IDEA (International Data Encryption Algorithm) được sử dụng để bảo vệ thông tin người dùng, bản mã; kết hợp với công nghệ chuỗi khối để bảo vệ một cách toàn diện nội dung tập tin. Sáng chế cũng đề xuất hệ thống thực hiện phương pháp này.

Hình 2.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề xuất phương pháp bảo vệ tập tin có entropy thông tin cao bằng hoán vị các thành phần dữ liệu của tập tin (mã hóa), sau đó, thông tin về việc hoán vị này (khóa mã) cùng với các thông tin liên quan đến người dùng sẽ được mã hóa bằng chuẩn mã hóa AES; đồng thời, tập sẽ được quy ước chia nhỏ thành các khối, các khối này được liên kết với nhau bằng mã một chiều SHA để kiểm soát sự toàn vẹn của tập tin. Mặt khác, toàn bộ thông tin về các sự kiện liên quan đến tập tin, từ sự kiện khởi phát (tạo ra tập tin) đến các sự kiện khác (như truy cập, sao chép, cấp quyền cho người dùng, thay đổi quyền người dùng…) sẽ được lưu trữ trên chính tập tin đó và khi có kết nối internet, sẽ đồng thời được lưu trữ tại máy chủ theo công nghệ chuỗi khối (block chain).

Bằng cách này, chủ sở hữu tập tin có thể hoàn toàn bảo vệ được nội dung tập tin, bao gồm kiểm soát việc truy cập để đọc dữ liệu của tập tin; sao chép, phân phối tập tin, ngăn chặn việc chỉnh sửa nội dung tập tin…; xóa, lựa chọn cấp quyền hoặc chặn truy cập vào tập tin đó đối với một/một số người khi có kết nối Internet.

Tình trạng kỹ thuật của sáng chế

Các tập tin hiện tại, đặc biệt là các tập tin âm thanh, video dân dụng và chuyên dụng thường có dung lượng lớn và chưa bảo mật nội dung. Các tập tin này chỉ đơn giản được tạo ra và lưu trữ trên thiết bị có tính năng nhớ dưới định dạng mở, tức là bất kỳ ai cũng dễ dàng đọc, sao chép. Việc cắt gọt, chỉnh sửa nội dung các tập tin loại này cũng có thể được thực hiện rất dễ dàng.

Trong một số trường hợp cần bảo mật, các dữ liệu này được bảo vệ bằng cách cất giữ theo quy trình bảo vệ dữ liệu, ví dụ như giữ trong két sắt, tuy nhiên như vậy vẫn tồn tại lỗ hổng bảo mật khi dữ liệu chưa được mã hóa kịp thời. Vì nhược điểm này, các nội dung riêng tư, nhạy cảm, quan trọng vẫn bị phát tán khi thiết bị lưu trữ bị thất lạc có thể gây hậu quả không mong muốn.

Một nhược điểm lớn nữa của các tập tin âm thanh, video được lưu trữ trong định dạng mở, không được bảo mật là chúng có thể dễ dàng bị cắt xén, chỉnh sửa, làm sai lệch nội dung và làm mất đi tính nguyên gốc. Do đó, các tập tin âm thanh, video luôn phải được cân nhắc và kiểm chứng về tính toàn vẹn của chúng trước khi sử dụng.

Để tăng cường bảo mật, việc phân phối video bằng đĩa CD hay thiết bị lưu trữ di động (như ổ cứng di động, USB) có thể bị thay thế bằng việc sử dụng một cơ sở dữ liệu tập trung. Tuy nhiên, cách làm đó lại loại bỏ tính thuận tiện của việc phân phối thông tin trên thiết bị lưu trữ di động và dữ liệu lưu trữ vẫn cần thêm một hệ thống mã hóa dữ liệu mạnh khác bảo vệ.

Để bảo mật đối với các tập tin âm thanh, video, có thể áp dụng các thuật toán mã hóa mạnh. Tuy nhiên, do đặc thù của các tập tin âm thanh, video thường có dung lượng lớn nên để mã hóa và giải mã đều yêu cầu phải có năng lực điện toán mạnh, dẫn đến giá thành và độ phức tạp của hệ thống mã hóa/giải mã tăng lên rất cao. Nếu năng lực điện toán không đủ mạnh thì việc giải mã (mở) tập tin sẽ cần nhiều thời gian, gây bất tiện cho người dùng.

Mã hoán vị đã được phát hiện và sử dụng từ lâu trong mã hóa thông tin. Mã hoán vị có nhiều cách làm khác nhau nhưng đều dựa trên một nguyên tắc chung là thay đổi vị trí các thành tố trong dữ liệu ban đầu một cách có kiểm soát, theo quy luật sao cho dữ liệu sau khi thay đổi vị trí trở nên không thể đọc được theo cách thông thường.

Nhược điểm của mã hoán vị là đối với những bản mã có entropy thông tin thấp, mã hoán vị dễ dàng bị giải mã bằng cách lợi dụng sự phân bố bất bình đẳng của các thành tố tạo nên dữ liệu. Theo đó, bên tấn công sẽ thống kê tần suất xuất hiện các thành tố trên bản mã, từ đó suy luận ra quy luật hoán vị (khóa mã). Ví dụ, theo các thống kê, trong các tập tin văn bản bằng tiếng Anh thì thành tố "e" xuất hiện nhiều nhất. Trong một tập tin được mã hóa là tập tin văn bản bằng tiếng Anh, có thể phá mã bằng cách tìm thành tố trong dữ liệu được mã hóa có tần suất xuất hiện cao nhất và gán cho nó là "e", và làm tiếp theo với các thành tố khác… Với năng lực điện toán hiện tại, các bản mã được mã hóa bằng mã hoán vị thông thường dễ dàng bị phá mã theo nguyên lý này.

Nhận thấy, các tập tin âm thanh, video hiện đại đều đã được lưu trữ theo các định dạng phổ biến hiện nay như MP3, MP4, JPEG…. Các tập tin này đều đã được trải qua quá trình nén entropy để làm giảm dung lượng; Các tập tin thuộc loại khác như tập tin văn bản, tập tin hình ảnh … cũng có thể được nén entropy bằng các phần mềm nén file như Winrar, Winzip với hiệu suất lớn và tốc độ xử lý cao. Quá trình nén entropy khiến cho sự phân bố của các thành tố tạo thành bản mã của các tập tin đã được xử lý nén entropy trở nên đồng đều (entropy thông tin cao). Xin tham chiếu tới *hình 1. Biểu đồ entropy thông tin của một tập tin video trước và sau khi nén entropy.* Do đó, việc bảo vệ các tập tin có entropy thông tin cao bằng mã hoán vị sẽ rất hiệu quả và nhanh chóng, đặc biệt đối với tập tin có dung lượng cao, trong khi vẫn loại trừ được nhược điểm bất bình đẳng của thành tố tạo nên dữ liệu, có thể bị khai thác để phá mã. Chỉ các thông tin có dung lượng thấp như thông tin về tên, mật khẩu của người dùng; thông tin về phân quyền; khóa mã … mới được bảo mật theo Chuẩn mã hóa AES (Advanced Encryption Standard).

Việc bảo vệ nội dung tập tin, đặc biệt là các tập tin âm thanh, video có dung lượng lớn, bằng cách kết hợp phương pháp hoán vị và chuẩn mã hóa AES sẽ khiến tốc độ mã hóa/giải mã tăng thêm nhiều lần so với việc chỉ áp dụng một phương pháp mã hóa (*bất đối xứng như RSA, đối xứng như AES*) trên cùng một năng lực điện toán. Bên cạnh đó, công nghệ chuỗi khối (block chain) và công nghệ mã một chiều SHA cũng được ứng dụng vào quá trình quản lý, phân quyền, sử dụng tập tin và ngăn chặn việc chỉnh sửa tập tin nhằm bảo vệ tập tin tốt hơn.

Vì lý do đó, sáng chế này đề xuất phương pháp bảo vệ nội dung tập tin có entropy thông tin cao bằng cách sử dụng kết hợp giữa mã hoán vị, chuẩn mã hóa AES và công nghệ chuỗi khối. Sáng chế cũng đề xuất hệ thống thực hiện phương pháp này.

Bản chất kỹ thuật của sáng chế

Mục đích của sáng chế là để bảo mật dữ liệu, giúp khắc phục những nhược điểm kỹ thuật của các phương pháp mã hóa dữ liệu trong tình trạng kỹ thuật đã biết.

Để đạt được mục đích nêu trên, sáng chế đề xuất phương pháp mã hóa dữ liệu, phương pháp lưu trữ lịch sử dữ liệu và sao chép dữ liệu, và phương pháp phân quyền người dùng có xác thực.

Phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao bằng cách sử dụng kết hợp mã hoán vị, chuẩn mã hóa AES và công nghệ chuỗi khối, gồm:

bước 1: mã hóa tập tin bằng mã hoán vị bằng cách sinh khóa mã (CypherKey) một cách ngẫu nhiên, sau đó mã hóa tập tin bằng mã hoán vị khóa mã (CypherKey) để tao ra bản mã theo công thức sau:

$$DataFile_{bản\ mã} = Swap(CypherKey, DataFile_{nội\ dung})$$

Trong đó $DataFile_{bản\ mã}$ là bản mã sau khi tập tin được mã hóa; $DataFile_{nội\ dung}$ là nội dung của tập tin khi chưa mã hóa, CypherKey là khóa mã, và Swap là hàm hoán vị.Theo một phương án của sáng chế, sau khi được mã hóa tại bước 1, tùy theo dung lượng, bản mã có thể được quy ước chia thành một hay nhiều khối; trên mỗi khối có một chữ ký số của khối được tính theo công thức:

$$Hash[j] = SHA(Hash[j-1], DataFile_{bản\ mã}, DataBlock[j])$$

bước 2: đặt mật khẩu để bảo vệ tập tin, hệ thống tính toán chữ ký số (hash) của mật khẩu của người dùng theo công thức:

Password_Hash= SHA(secretkey, username, password, permission)

Trong đó, secret-key là khóa bí mật, riêng có của hệ thống và được ẩn đi, username là tên người dùng, password là mật khẩu do người dùng thiết lập và permission là quyền của người dùng đối với tập tin. Việc đặt mật khẩu có thể được tiến hành bởi người dùng, hoặc người quản trị, tùy thuộc việc phân cấp cho tác vụ này. Chữ ký số của mật khẩu của người dùng được lưu trữ một cách công khai, được sử dụng để đối chiếu với mật khẩu người dùng.

bước 3: mã hóa khóa mã và thông tin tài khoản người dùng bằng chuẩn mã hóa đối xứng.

Theo một khía cạnh của sáng chế, chuẩn mã hóa đối xứng có thể được chọn chuẩn bất kỳ trong số các chuẩn mã hóa đối xứng bao gồm Twofish, Serpent, AES (còn được gọi là Rijndael), Blowfish, CAST5, RC4, Tam phần DES (Triple DES), và IDEA (International Data Encryption Algorithm), v.v.

Theo một phương án thực hiện của sáng chế, phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao theo sáng chế có chuẩn mã hóa đối xứng là AES, trong đó:

- Thông tin tài khoản người dùng được mã hóa theo công thức:

ProtectedUserInfo = AES.Encrypt(secretkey, username, password, permission)

- Mỗi người sẽ có một Khóa mã (CypherKey) được mã hóa riêng theo công thức:

ProtectedCypherKey=AES.Encrypt(secretkey, password, CypherKey)

Theo một phương án thực hiện của sáng chế, phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao theo sáng chế còn bao gồm bước 4 và bước 5 sau bước 3.

bước 4: Tạo ra tập tin "thùng chứa" (container file) để chứa toàn bộ các thông tin của tập tin đã được mã hóa sẽ . Tập tin thùng chứa bao gồm: bản mã ($datafile_{bản\ mã}$); chữ ký số của mật khẩu của tất cả những người/nhóm người dùng đã được cấp phép (password_hash) và thông tin đã được mã hóa về tất cả các người dùng/nhóm người dùng (protected UserInfo) và tất cả các khóa mã đã được mã hóa (protectedCypherkey).

bước 5: Tạo ra bảng dữ liệu sự kiện, lưu trữ toàn bộ lịch sử của tập tin từ lúc được tạo ra (khởi phát) và các sự kiện xảy ra với đến tập tin đó sau thời điểm khởi phát theo công nghệ chuỗi khối, và lưu trữ bảng dữ liệu sự kiện này trên một phương tiện lưu trữ tập tin như là một phần mở rộng của tập tin thùng chứa hoặc như một tập tin riêng.

Khối khởi phát và các khối tiếp theo trong chuỗi khối sẽ được đóng dấu theo công thức: Block_Seal= SHA[block_Seal khối liền trước; thông tin sự kiện]

Nếu là khối khởi phát thì tham số "block_seal khối liền trước" được đặt bằng giá trị nul;

Nếu người dùng sao chép một tập tin thì một khối mới trên tập tin gốc sẽ được tạo ra để ghi nhận sự kiện này; đồng thời, trên tập tin mới được tạo ra thì sự kiện sao chép sẽ được coi là sự kiện khởi phát của tập tin mới được tạo ra đó.

Theo một phương án thực hiện, phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao còn bao gồm bước 6 là đồng bộ hóa bảng dữ liệu sự kiện tại phương tiện lưu trữ tập tin tại bước 5 với bảng dữ liệu sự kiện trên hệ thống máy chủ kết nối internet. Trong đó, trên máy chủ có chứa Bảng dữ liệu sự kiện toàn cầu, chứa tất cả các thông tin liên quan đến tập tin, theo đó, thông tin sự kiện đối với tập tin gốc và tất cả các tập tin được sao chép từ tập tin gốc được lưu trữ trên tất cả các Phương tiện lưu trữ tập tin sẽ được cập nhật trên Bảng dữ liệu sự kiện toàn cầu thông qua hệ thống internet. Ứng với mỗi sự kiện sẽ có một khối mới được tạo ra, cùng với một con dấu khối toàn cầu. Con dấu khối toàn cầu được tính theo công thức:

Block_Seal Global= SHA[block_Seal khối liền trước; thông tin sự kiện]

Trên Bảng dữ liệu sự kiện tại Phương tiện lưu trữ tập tin có thêm trường thông tin ghi nhận con dấu khối toàn cầu (block_seal_global); theo đó, khi thông tin sự kiện được cập nhật với hệ thống toàn cầu, một con dấu khối toàn cầu được tạo ra và gửi về Phương tiện lưu trữ tập tin để lưu trữ.

Trong một phương án ưu tiên của sáng chế, ngoài các thông tin sự kiện liên quan đến tập tin, tập tin gốc đồng thời được lưu trữ trên hệ thống máy chủ, cho phép người dùng thực hiện quyền của mình đối với tập tin một cách trực tiếp trên máy chủ, gồm tìm kiếm, truy cập để xem, tải về phương tiện lưu trữ, xóa hoặc cấp quyền cho người dùng và các quyền khác.

Trong một phương án của sáng chế, việc mã hóa tập tin bằng mã hoán vị (bước 1) được thực hiện bằng cách hoán vị các tiểu phân đoạn trong từng phân đoạn của tập tin dựa trên khóa mã, lần lượt theo quy trình sau:

(i) Phân chia tập tin thành các phân đoạn và các tiểu phân đoạn: theo đó, tập tin sẽ được phân chia thành nhiều phân đoạn bằng nhau, giá trị của mỗi phân đoạn là $2^n \times 256$ byte, trong đó n là số nguyên lớn hơn hoặc bằng không. Trong trường hợp việc chia nhỏ tập tin thành các phân đoạn với giá trị bằng nhau là $2^n \times 256$ byte có số dư thì phần dư sẽ được coi là một phân đoạn độc lập riêng và phần phân đoạn này không được mã hóa.

Mỗi phân đoạn sẽ được chia thành 256 tiểu phân đoạn bằng nhau, mỗi tiểu phân đoạn sẽ có giá trị tương ứng bằng 2^n byte, trong đó n là số nguyên không âm.

(ii) Một khóa mã (Cypherkey) mô tả trình tự sắp xếp các tiểu phân đoạn với độ dài 256 phần tử được tạo ra một cách ngẫu nhiên;

(iii) Hoán vị trình tự tiểu phân đoạn trong tất cả các phân đoạn, từ trình tự thông thường (lần lượt từ 0 đến 255) sang trình tự do khóa mã quy định; việc hoán vị trình tự các tiểu phân đoạn này được thực hiện một cách thống nhất trong tất cả các phân đoạn của tập tin.

Trong một phương án ưu tiên của sáng chế, tham số n có giá trị từ 2 đến 4.

Trong một phương án của sáng chế, việc mã hóa tập tin bằng mã hoán vị (bước 1) được thực hiện bằng cách hoán vị từ mã, lần lượt theo quy trình sau:

(i) một khóa mã mô tả trình tự sắp xếp các từ mã có số lượng phần tử bằng với số lượng từ mã sẽ được tạo ra một cách ngẫu nhiên,

(ii) từ mã trên tập tin được hoán vị thành từ mã tương ứng do Cypherkey quy định theo nguyên tắc hoán vị từng từ mã.

Trong một phương án ưu tiên của sáng chế, số lượng từ mã là 256.

Trong một phương án, phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao theo sáng chế này có thêm bước 1b ngay sau bước 1. Theo đó, sau khi được mã hóa tại bước 1, tùy theo dung lượng, bản mã có thể được quy ước chia thành một hay nhiều khối; và gãn mỗi khối cho một chữ ký số của khối, được tính theo công thức:

Hash[j] = SHA(Hash[j-1], DataFile,DataBlock[j])

Trong đó, SHA là hàm mã hóa 1 chiều; Hash[j-1] là chữ ký số của khối liền trước; nếu bản mã chỉ có duy nhất một khối hoặc là khối đầu tiên thì Hash[j-1] sẽ có giá trị nul; DataFile là tập tin sau khi đã được mã hóa (bản mã); DataBlock[j] là khối thứ j trong dữ liệu. Nếu bản mã chỉ có duy nhất một khối hoặc là khối đầu tiên, thì tham số DataBlock sẽ bằng với giá trị bản mã (DataFile).

Trong một phương án của sáng chế, mỗi khối được quy ước có dung lượng từ 1MB đến 16MB.

Sáng chế cũng đề xuất hệ thống thực hiện phương pháp bảo vệ nội dung tập tin bao gồm: thiết bị lưu trữ và nén entropy cho tập tin ; thiết bị dùng để mã hóa/giải mã tập tin đã nén entropy ở thiết bị lưu trữ và tính toán các tham số theo phương pháp của sáng chế này; máy chủ có kết nối internet để lưu trữ thông tin về tập tin, từ sự kiện khởi phát đến các sự kiện khác có liên quan đến tập tin được lưu trữ trên thiết bị lưu trữ tập tin.

Trong một phương án của sáng chế, hệ thống thực hiện phương pháp bảo vệ nội dung tập tin, khác biệt ở chỗ, máy chủ có kết nối internet có khả năng lưu trữ thêm tập tin gốc.

Mô tả vắn tắt các hình vẽ

Hình 1 là biểu đồ entropy thông tin của một tập tin video trước và sau khi nén entropy. Công thức tính Entropy thông tin ở đây là công thức của Claude Shannon:

$$H(x) = -\sum_{i=1}^{n} p(i) \log_2 p(i)$$

Với p(i) là xác suất xảy ra của giá trị i. Entropy của x cũng là giá trị kì vọng của các độ ngạc nhiên của các giá trị mà x có thể nhận. Entropy thông tin trong trường hợp phần tử tín hiệu ngẫu nhiên rời rạc còn được gọi là entropy Shannon.

Hình 2 là lưu đồ minh họa cách thức hoạt động cơ bản của phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao.

Mô tả chi tiết sáng chế

Sau đây, sáng chế sẽ được mô tả chi tiết với các phương án. Phương án thực hiện sáng chế sau đây được đưa ra làm ví dụ nhằm mục đích bộc lộ toàn bộ sáng chế với người có hiểu biết trung bình trong lĩnh vực kỹ thuật tương ứng. Tuy nhiên, các ví dụ không làm giới hạn sáng chế. Sáng chế bao gồm tất cả các biến thể, các phép tương đương và các lựa chọn thay thế thuộc tinh thần và phạm vi của sáng chế.

Cần phải hiểu rằng, nếu không có chỉ dẫn khác, các thuật ngữ được dùng trong bản mô tả nên được hiểu là đã được người có hiểu biết trung bình về lĩnh vực kỹ thuật này hiểu rõ và sử dụng rộng rãi. Các thuật ngữ được sử dụng trong bản mô tả sáng chế là nhằm mục đích mô tả các phương án cụ thể và không phải là để giới hạn. Các thuật ngữ như Cypherkey, Onx, Ony, Onz,… được dùng để phân biệt các đối tượng, tham số, hàm số, thuật toán và không làm giới hạn sáng chế. Các mô tả đã biết hoặc có thể làm lu mờ các điểm quan trọng của sáng chế sẽ được bỏ qua.

Trong sáng chế này, các thuật ngữ sau đây được hiểu như sau:

1. Từ mã: Từ mã là ký hiệu mang theo thông tin. Ví dụ, chữ "Nhật" trong tiếng Trung có hình chữ nhật dọc và nét ngang ở giữa (日) mang thông tin là khái niệm Mặt trời; ký tự "a" mang theo thông tin là nguyên âm "a".

Trong máy tính từ mã là 1 số gồm n bít, ví dụ n=8 bít hay 1 byte, n còn được gọi là độ dài của từ mã, ví dụ số 3 có 8 bít được biểu diễn trong các hệ cơ số là 00000011|bin hoặc 0x03|hex hoặc 3|dec và tự hiểu là 8 bít. Trong điều kiện bình thường (không mã hóa), số nào mang thông tin là số đó, ví dụ 00000011|bin là từ mã, mang thông tin là số 3.

Trong phép hoán vị từ mã, từ mã 00000011|bin được hoán đổi mang thông tin khác, khiến cho việc đọc tập tin theo cách thông thường trở nên không thể;

Các từ mã có thể có độ dài khác nhau, nhưng trong phạm vi sáng chế này chỉ xem xét các từ mã có độ dài bằng nhau.

2. Không gian từ mã: Tập hợp các từ mã có trong tập tin tạo thành Không gian từ mã. Không gian từ mã n-bit là tập hợp tất cả các từ mã n-bit có thể có, bao gồm 2^n từ mã khác nhau, ký hiệu là Vn

3. Phân đoạn: tập tin được quy ước chia ra thành nhiều phần bằng nhau có giá trị mỗi phần bằng $2^n \times 256$ byte, gọi là phân đoạn. Tuy nhiên, việc phân chia chỉ là quy ước, tập tin vẫn toàn vẹn, không bị phân chia về mặt vật lý.

Trường hợp dung lượng của tập tin không chia hết cho giá trị của phân đoạn thì phần còn thừa sẽ là một phân đoạn độc lập, được giữ nguyên, không xáo trộn theo trình tự Cypherkey.

4. Khối: Khối là phần quy ước của tập tin, nhưng khái niệm "khối" không đồng nhất với khái niệm "phân đoạn" nêu trên vì chức năng của chúng khác nhau.

Tập tin được quy ước chia thành nhiều khối bằng nhau với giá trị định trước. Tập tin được quy ước chia thành các khối nhưng vẫn toàn vẹn về mặt vật lý. Các khối được liên kết với nhau bằng mã một chiều SHA, có tác dụng kiểm soát sự toàn vẹn của tập tin.

5. Khóa mã (còn gọi là Cypherkey): là trình tự xáo trộn dữ liệu, được áp dụng thống nhất cho các phân đoạn của tập tin hoặc cho các từ mã của tập tin. Nó là một dãy số gồm N số có giá trị từ 0 đến (N-1); nhưng không tuần tự từ 0 đến (N-1) mà theo trình tự ngẫu nhiên.

6. Bản mã: là tập tin sau khi các từ mã/tiểu phân đoạn đã được hoán vị (mã hóa). Bản mã được bộc lộ công khai và phơi nhiễm cho việc tấn công nhưng không thể hiểu được nếu không có Cypherkey. Bản mã theo sáng chế này giữ nguyên đặc tính entropy thông tin cao của tập tin ban đầu, tức là việc phân bố của các từ mã trong tập tin có tính đồng đẳng nên hầu như không thể giải mã dựa trên việc phân tích sự bất bình đẳng của các thành tố của tập tin.

7. Sự kiện khởi phát: là sự kiện tạo lập nên tập tin, gồm thông tin về thời gian, địa điểm, tác giả tạo ra tập tin; những người được sử dụng tập tin và quyền của họ.

8. Bảng dữ liệu sự kiện: là một bảng chứa các thông tin liên quan đến tập tin, bao gồm nhưng không giới hạn bởi các thông tin như chữ ký số của người dùng; mật khẩu đã mã hóa; thông tin người dùng đã được mã hóa … Bảng được cập nhật thường xuyên khi có các sự kiện xảy ra với tập tin ấy.

Bảng dữ liệu sự kiện có thể tồn tại dưới dạng một tập tin mở rộng nằm trong tập tin thùng chứa, hoặc dưới dạng một tập tin đính kèm bên ngoài tập tin thùng chứa.

Bảng dữ liệu sự kiện được lưu trữ trên Thiết bị Lưu trữ tập tin. Trong một phương án ưu tiên, Bảng dữ liệu sự kiện được lưu trữ đồng thời trên thiết bị lưu trữ tập tin và trên máy chủ; theo đó, Bảng dữ liệu sự kiện trên Thiết bị lưu trữ tập tin chứa các thông tin sự kiện liên quan đến tập tin trên Thiết bị đó; Bảng dữ liệu sự kiện trên máy chủ, gọi là Bảng dữ liệu sự kiện toàn cầu, chứa toàn bộ thông tin sự kiện liên quan đến tập tin gốc (tập tin khởi phát) và các tập tin được sao chép từ tập tin khởi phát.

9. Tập tin có entropy cao: Tập tin có entropy cao được mã hóa theo sáng chế này là các tập tin âm thanh, video và các tập tin chứa dữ liệu khác đã được nén entropy. Việc nén entropy đối với tập tin âm thanh, video hiện đại thường đã được thực hiện bằng các phần mềm phổ biến theo các định dạng thông thường như MP3, MP4, JPEG… để giảm dung lượng; các tập tin dữ liệu khác như tập tin văn bản, hình ảnh … cũng có thể được nén entropy bằng các phần mềm nén dữ liệu phổ biến như Winrar, Winzip. Qua đó, sự bất bình đẳng của các thành tố tạo thành dữ liệu hầu như đã được loại trừ.

Như các chuyên gia trong lĩnh vực đã biết, để bảo vệ các tập tin âm thanh, video, văn bản, hình ảnh v.v. có entropy thông tin thấp, mã hoán vị dễ dàng bị giải mã bằng cách lợi dụng sự phân bố bất bình đẳng của các thành tố tạo nên dữ liệu. Do đó, việc tấn công bằng cách thống kê tần suất xuất hiện các thành tố trên bản mã suy luận ra quy luật hoán vị (khóa mã) sẽ khá sẽ dễ dàng.

Tuy nhiên, tác giả sáng chế phát hiện ra rằng đối với các tập như video, hình ảnh, âm thanh, văn bản v.v. đã trải qua quá trình nén entropy để làm giảm dung lượng như MP3, MP4, JPEG, Winrar, Winzip v.v., thì quá trình nén entropy này khiến cho sự phân bố của các thành tố tạo thành bản mã của các tập tin đã được xử lý nén entropy trở nên đồng đều (được gọi là entropy thông tin cao). Chính vì vậy, nếu kết hợp việc tiến hành bảo mật các tập tin có entropy thông tin cao này bằng mã hoán vị sẽ rất hiệu quả và nhanh chóng, đặc biệt đối với tập tin có dung lượng cao, trong khi vẫn loại trừ được nhược điểm bất bình đẳng của thành tố tạo nên dữ liệu nêu trên. Hơn nữa, nếu tiến hành kết hợp mã hoán vị với chuẩn mã hóa AES và công nghệ chuỗi khối trong việc bảo mật các tập tin này thì chúng sẽ được bảo mật ở mức rất cao, và hầu như không thể tấn công được với công nghệ hiện tại.

Với các phát hiện ở trên, sáng chế này được thực hiện thông qua các thành phần chính như sau:

(i) Thiết bị lưu trữ tập tin, có khả năng nén entropy đối với tập tin.

Thiết bị lưu trữ tập tin có thể là thiết bị có chức năng ghi âm (với tập tin âm thanh),

chức năng ghi hình (với tập tin hình ảnh/video), hoặc máy tính có khả năng tạo lập các loại tập tin khác. Các tập tin được tạo lập phải được xử lý bằng các phần mềm nén dữ liệu, làm cho entropy thông tin của tập tin trở nên cao. Các định dạng nén dữ liệu phổ thông hiện nay có thể kể đến định dạng MP3, MP4, JPEG hay Winrar, Winzip …

Thiết bị lưu trữ tập tin có thể là các máy ghi âm, chụp ảnh, ghi hình, điện thoại thông minh có các chức năng nêu trên được sử dụng phổ biến hoặc các máy tính thông thường.

(ii) Thiết bị có khả năng mã hóa/giải mã tập tin theo phương pháp để xuất theo sáng chế này. Thiết bị này có thể là thiết bị trộn (mixer) riêng hoặc mô đun hoặc phần mềm tích hợp trong Thiết bị lưu trữ tập tin.

Trong một phương án, hệ thống có thêm (iii) Máy chủ (server) kết nối internet có khả năng lưu trữ thông tin về tập tin, từ sự kiện khởi phát đến các sự kiện khác có liên quan đến tập tin. Theo đó, khi tập tin được tạo ra, các thông tin về sự kiện khởi phát và các thông tin sự kiện khác (đọc, sao chép, chia sẻ, cấp quyền cho người dùng, thay đổi thông tin người dùng …) sẽ được gửi về máy chủ; từng khối dữ liệu sẽ được máy chủ đóng dấu và lưu trữ trên máy chủ. Bằng cách đó, hệ thống có thể truy vết sự kiện liên quan đến tập tin ngay khi thiết bị đang lưu trữ tập tin đó có kết nối internet. Tuy nhiên, hệ thống không biết được nội dung tập tin là gì.

Trong một phương án ưu tiên, hệ thống bao gồm máy chủ kết nối internet có khả năng lưu trữ thông tin về tập tin, từ sự kiện khởi phát đến các sự kiện khác có liên quan đến tập tin. Khác biệt ở chỗ, khi tập tin được tạo ra, toàn bộ tập tin cùng các thông tin về sự kiện khởi phát và các thông tin sự kiện khác (đọc, sao chép, chia sẻ, cấp quyền cho người dùng, thay đổi thông tin người dùng …) sẽ được đồng thời gửi về và lưu trữ trên máy chủ; từng khối dữ liệu sẽ được máy chủ đóng dấu và lưu trữ trên máy chủ, bên cạnh việc đóng dấu và lưu trữ trên Thiết bị lưu trữ tập tin. Khi đó, dữ liệu về tập tin trên máy chủ là đầy đủ, và máy chủ có quyền cao nhất với tập tin đó.

Trong phương án ưu tiên này, máy chủ thực hiện việc lưu trữ, mã hóa/giải mã dữ liệu; quản lý nội dung (CMS), gồm tra cứu thông tin, xem thông tin trên trình duyệt web, tải tập tin về máy, đăng ký người dùng, phân quyền cho người dùng); và lưu trữ toàn bộ sự kiện liên quan đến tập tin, từ sự kiện khởi phát đến các sự kiện khác như đọc, sao chép, chia sẻ …

Bằng cách đó, hệ thống có thể kiểm soát một cách thống nhất và toàn diện về mọi khía cạnh liên quan đến việc lưu trữ, sử dụng, phát tán tập tin.

Theo phương pháp đề xuất theo sáng chế này, tập tin có entropy cao được bảo vệ theo các bước sau đây:

Bước 1. Mã hóa tập tin bằng mã hoán vị

Trong một phương án của sáng chế, đầu tiên khóa mã (Cypher) được hệ thống sinh ra một cách ngẫu nhiên, sau đó tập tin được mã hóa bằng cách hoán vị các tiểu phân đoạn trong từng phân đoạn của tập tin dựa trên khóa mã (CypherKey) để tao ra bản mã theo công thức sau:

$$DataFile_{bản\ mã} = Swap(CypherKey, DataFile_{nội\ dung})$$

Trong đó $DataFile_{bản\ mã}$ là bản mã sau khi tập tin được mã hóa; $DataFile_{nội\ dung}$ là nội dung của tập tin khi chưa mã hóa, CypherKey là khóa mã, và Swap là hàm hoán vị.

Theo phương án này, bước mã hóa tập tin sẽ được thực hiện như sau:

(i) Phân chia tập tin thành các phân đoạn và các tiểu phân đoạn: theo đó, tập tin sẽ được quy ước chia nhỏ thành nhiều phân đoạn bằng nhau, giá trị của mỗi phân đoạn là $2^n \times 256$ byte, trong đó n là số nguyên lớn hơn hoặc bằng không. Trong trường hợp việc chia nhỏ tập tin thành các phân đoạn với giá trị bằng nhau là $2^n \times 256$ byte có số dư thì phần dư sẽ được coi là một phân đoạn độc lập riêng và phần phân đoạn này không được mã hóa.

Trong một phương án ưu tiên, tham số "n" có giá trị từ 2 đến 4.

Mỗi phân đoạn sẽ được chia thành 256 tiểu phân đoạn bằng nhau, mỗi tiểu phân đoạn sẽ có giá trị tương ứng bằng 2^n byte.

Cần lưu ý là việc phân chia tập tin thành các phân đoạn chỉ mang tính quy ước để phục vụ việc hoán vị các tiểu phân đoạn chứ không phân chia về mặt vật lý. Tức là, về bản chất, tập tin vẫn giữ nguyên trạng, không hề bị chia tách thành các phân đoạn. Mã hoán vị là bất kỳ cách thức hoán vị nào sử dụng khóa mã Cypherkey để tiến hành hoán vị các tiểu phân đoạn, và khóa mã Cypherkey mô tả quy luật hoặc trình tự hoán vị các tiểu phân đoạn, và dựa vào đó để khôi phục lại tập tin gốc.

(ii) Một khóa mã (Cypherkey) mô tả trình tự sắp xếp các tiểu phân đoạn với độ dài 256 phần tử sẽ được tạo ra một cách ngẫu nhiên.

(iii) Hoán vị trình tự tiểu phân đoạn trong các phân đoạn, từ trình tự thông thường *(lần lượt từ 0 đến 255)* sang trình tự do khóa mã quy định. Việc hoán vị trình tự các tiểu phân đoạn này được thực hiện một cách thống nhất trong tất cả các phân đoạn của tập tin.

Vì số lượng tiểu phân đoạn trong mỗi phân đoạn là 256 thì quá lớn để mô tả nên để đơn giản hóa, có thể xem xét một ví dụ với một phân đoạn có số lượng tiểu phân đoạn giả thiết là 4. Theo đó, các tiểu phân đoạn theo trình tự thông thường trong các phân đoạn lần lượt là N=4:[1,2,3,4].

Trong trường hợp này, khóa mã (Cypherkey) với độ dài bằng 4 sẽ được biến đổi thành trình tự mới mang tính ngẫu nhiên, giả sử như biến 4 => 2, 1 => 4, 2 => 3, 3 => 1; khi đó,

Cypherkey sẽ có dạng: N=4:[4,3,1,2]. Áp dụng khóa mã này vào việc mã hóa sẽ được thực hiện bằng cách: mỗi vị trí của tiểu phân đoạn cũ được thay bằng một tiểu phân đoạn mới do khóa mã quy định, ví dụ với dữ liệu mà các phân đoạn có trình tự tiểu phân đoạn là [1,2,3,4] sẽ được mã hóa thành dữ liệu mà các phân đoạn có trình tự tiểu phân đoạn là [4,3,1,2].

Cần lưu ý là việc tạo ra Cypherkey là hoàn toàn ngẫu nhiên, không theo một quy luật nào. Nếu độ dài của khóa mã là 4 thì việc phá mã khá đơn giản, vì chỉ cần thử nghiệm số biến bằng giai thừa của 4 (4!), tức là bằng: 4×3×2=24 biến. Tuy nhiên, khi độ dài của khóa mã N=256 thì số lượng biến phải thử nghiệm bằng giai thừa của 256 (256!), tức là một con số vô cùng lớn đến mức năng lực điện toán hiện tại không thể xử lý được.

Trong một phương án khác của sáng chế, việc mã hóa tập tin sẽ được thực hiện bằng việc hoán vị từ mã.

Thông tin của mỗi tập tin được thể hiện bằng nhiều từ mã. Trong các tập tin âm thanh, hình ảnh được nén theo các chuẩn phổ biến hiện nay, số lượng từ mã được sử dụng thường là 256. Tuy nhiên, số lượng từ mã được sử dụng để thể hiện thông tin trong tập tin không bị giới hạn ở con số 256.

Theo phương án này, bước mã hóa tập tin sẽ được thực hiện như sau:

(i) Một khóa mã (Cypherkey) mô tả trình tự sắp xếp các từ mã có số lượng phần tử bằng với số lượng từ mã sẽ được tạo ra một cách ngẫu nhiên.

(ii) Hoán vị từ mã theo nguyên tắc hoán vị từng từ mã, theo đó, khi hệ thống gặp một từ mã trong tập tin ban đầu thì sẽ hoán vị ngay thành từ mã tương ứng do Cypherkey quy định.

Để đơn giản hóa, có thể xem xét một ví dụ với tập tin có số lượng từ mã là 4, gồm các từ mã là a, b, c, d.

Giả thiết khóa mã với độ dài bằng 4 sẽ được biến đổi thành trình tự mới mang tính ngẫu nhiên, giả sử như biến a => b, b => c, c => d, d => a; khi đó, Cypherkey sẽ có dạng: N=4:[b,c,d,a]. Áp dụng khóa mã này vào việc mã hóa sẽ được thực hiện bằng cách với mỗi từ mã cũ được thay bằng từ mã mới do khóa mã quy định, ví dụ với dữ liệu [a,a,b,b,c,d,d] sẽ được mã hóa thành [b,b,c,c,d,a,a].

Việc tạo ra Cypherkey là hoàn toàn ngẫu nhiên, không theo một quy luật nào. Nếu độ dài của khóa mã là 4 thì việc phá mã khá đơn giản, vì chỉ cần thử nghiệm số biến bằng giai thừa của 4 (4!), tức là bằng: 4×3×2=24 biến. Tuy nhiên, khi độ dài của khóa mã N=256, tức là số lượng từ mã đang được sử dụng phổ biến hiện nay trong các tập tin âm thanh, hình ảnh,

video thì số lượng biến phải thử nghiệm bằng giai thừa của 256 (256!), tức là một con số vô cùng lớn đến mức năng lực điện toán hiện tại không thể xử lý được.

Trong một phương án ưu tiên, quy trình bảo vệ nội dung tập tin có thêm bước 1b. Theo đó, tập tin sau khi được mã hóa (bản mã) được quy ước chia thành nhiều khối, trên mỗi khối có chữ ký số nhằm đảm bảo tính toàn vẹn của dữ liệu, cụ thể như sau:

Bước 1b. Sau khi được mã hóa, Bản mã được ký bằng chữ ký số tạo bởi mã một chiều SHA để đảm bảo sự toàn vẹn. Theo đó, tùy theo dung lượng, bản mã có thể được quy ước chia thành một hay nhiều khối; mỗi khối có dung lượng tối ưu nằm trong khoảng từ 1MB đến 16MB bởi vì nếu dung lượng của khối quá lớn thì công năng của việc chia khối không cao, dung lượng khối quá nhỏ sẽ làm tăng số lượng chữ ký số (HASH), dẫn đến làm tăng dung lượng lưu trữ.

Chữ ký số (Hash) của từng khối của bản mã sẽ được tính theo công thức:

Hash[j] = SHA(Hash[j-1], DataFile,DataBlock[j])

Trong đó,

SHA là hàm mã hóa 1 chiều, có thể là SHA1, SHA2, SHA3, Keccak, Shake, hay các dạng tương tự

Hash[j-1] là hash của block liền trước

DataFile là tập tin sau khi đã được mã hóa (bản mã)

DataBlock[j] là khối thứ j trong dữ liệu

Trong trường hợp bản mã chỉ có duy nhất một khối hoặc là khối đầu tiên thì Hash[j-1] sẽ có giá trị nul; DataBlock sẽ bằng với giá trị bản mã (DataFile).

Bước 2: đặt mật khẩu người dùng

Khi người dùng đặt mật khẩu bảo vệ tập tin, hệ thống tính toán chữ ký số (hash) của mật khẩu của người dùng theo công thức:

Password_Hash= SHA(secretkey, username, password, permission)

Trong đó,

- *secret-key là khóa bí mật, riêng có của hệ thống và được ẩn đi.*
- *username là tên người dùng;*
- *password là mật khẩu do người dùng thiết lập;*
- *permission là quyền của người dùng đối với tập tin, được quy ước mỗi quyền là một số tự nhiên từ 0 trở lên, ví dụ trong trường hợp quy định người dùng có 4*

quyền đối với tập tin thì quyền đọc được quy ước là 1, 2 là sao chép, 3 là chia sẻ và 4 là quyền xóa tập tin...

Chữ ký số của mật khẩu của người dùng được lưu trữ một cách công khai, được sử dụng để đối chiếu với mật khẩu người dùng thay vì lưu trữ mật khẩu người dùng.

Bước 3: mã hóa khóa mã, thông tin tài khoản người dùng bằng chuẩn mã hóa hai chiều đối xứng AES.

Trong sáng chế này, toàn bộ khóa mã, thông tin tài khoản người dùng được mã hóa bằng chuẩn mã hai chiều đối xứng AES, theo đó:

- Thông tin tài khoản người dùng được mã hóa theo công thức:

ProtectedUserInfo = AES.Encrypt(secretkey, username, password, permission)

- Mỗi người sẽ có một Khóa mã (CypherKey) được mã hóa riêng theo công thức:

protectedCypherKey=AES.Encrypt(secretkey, password, CypherKey)

Cần lưu ý rằng, mặc dù sáng chế bộc lộ việc mã hóa khóa mã, thông tin tài khoản người dùng bằng chuẩn mã hóa hai chiều đối xứng AES, song chuẩn mã hóa hai chiều đối xứng AES được đưa ra chỉ nhằm làm ví dụ minh họa cho sáng chế. Chuyên gia trung bình trong lĩnh vực kỹ thuật hoàn toàn có thể thực hiện bằng các chuẩn mã hóa khác. Vì vậy, sáng chế không bị giới hạn vào chuẩn mã hóa hai chiều đối xứng AES, mã cso thể tiến hành với các loại chuẩn mã hóa đối xứng khác, chẳng hạn Twofish, Serpent, Blowfish, CAST5, RC4, Tam phần DES (Triple DES), và IDEA (International Data Encryption Algorithm etc.

Bước 4: một tập tin "thùng chứa" (container file), chứa toàn bộ các thông tin của tập tin đã được mã hóa sẽ được tạo ra; tập tin thùng chứa sẽ bao gồm các thông tin sau đây:

(i) Bản mã (datafile);

(ii) Chữ ký số của mật khẩu của tất cả những người/nhóm người dùng đã được cấp phép (password_hash).

(iii) Thông tin đã được mã hóa về tất cả các người dùng/nhóm người dùng (protected UserInfo) và tất cả các khóa mã (protectedCypherkey).

Bước 5: Bảng dữ liệu sự kiện, lưu trữ toàn bộ lịch sử của tập tin từ lúc được tạo ra (khởi phát) và các sự kiện xảy ra với đến tập tin đó sau thời điểm khởi phát theo công nghệ chuỗi khối (block-chain), sẽ được tạo ra và lưu trữ như là một phần mở rộng của tập tin thùng chứa hoặc như một tập tin riêng.

Theo đó, khối đầu tiên của chuỗi khối là thông tin sự kiện khởi phát, gồm thông tin về thời gian, địa điểm, tác giả tạo ra tập tin; những người được sử dụng tập tin và quyền của họ.

Khi có sự kiện mới xảy ra với tập tin, gồm truy cập để đọc, sao chép; thay đổi người dùng; thay đổi quyền của người dùng; xóa tập tin… thì một khối mới được tạo ra trên Bảng dữ liệu sự kiện để ghi nhận lại các sự kiện này.

Khối khởi phát và các khối tiếp theo trong chuỗi khối sẽ được đóng dấu theo công thức:

Block_Seal= SHA[block_Seal khối liền trước; thông tin sự kiện]

Trong trường hợp Block là khối khởi phát thì tham số "block_seal khối liền trước" được đặt bằng giá trị nul.

Trong trường hợp người dùng sao chép một tập tin thì một khối mới trên tập tin gốc sẽ được tạo ra để ghi nhận sự kiện này; đồng thời, trên tập tin mới được tạo ra thì sự kiện sao chép sẽ được coi là sự kiện khởi phát của tập tin mới được tạo ra đó.

Kết thúc quy trình, toàn bộ nội dung tập tin có entropy thông tin cao ban đầu được mã hóa bằng việc hoán vị các từ mã hoặc hoán vị các tiểu phân đoạn trong các phân đoạn của tập tin để tạo thành một bản mã (DataFile) sẵn sàng phơi nhiễm cho việc tấn công; chữ ký số của mật khẩu của người dùng được tạo ra bằng cách mã hóa bằng mã một chiều SHA; thông tin người dùng và khóa mã được bảo vệ bằng chuẩn mã hóa AES; các khối của bản mã được đóng dấu bằng mã một chiều SHA để đảm bảo sự toàn vẹn của dữ liệu.

Bản mã, khóa mã, chữ ký số của mật khẩu của người dùng; thông tin tài khoản người dùng và con dấu của các khối được mã hóa bằng chuẩn mã hóa hai chiều đối xứng AES sẽ được lưu trữ trong một tập tin "thùng chứa", trong đó, sự kiện khởi phát và các sự kiện xảy ra với tập tin sẽ được lưu trữ trong Bảng dữ liệu sự kiện thành các khối (block) thuộc chuỗi khối (block chain), được đóng dấu bằng mã một chiều SHA.

Trong một phương án ưu tiên, Bảng dữ liệu sự kiện tại Phương tiện lưu trữ tập tin được đồng bộ hóa với Bảng dữ liệu sự kiện trên Hệ thống máy chủ kết nối internet tại Bước 6.

Bước 6: Bảng dữ liệu sự kiện tại Phương tiện lưu trữ tập tin được đồng bộ hóa với Bảng dữ liệu sự kiện trên hệ thống máy chủ kết nối Internet. Theo đó,

- Trên Bảng dữ liệu sự kiện tại Phương tiện lưu trữ tập tin có thêm trường thông tin ghi nhận con dấu khối toàn cầu (block_seal_global). Khi sự kiện xảy ra với tập tin lưu trữ tại Phương tiện lưu trữ tập tin được cập nhật lên hệ thống máy chủ kết nối Internet, máy chủ sẽ tính toán con dấu của khối toàn cầu (Block_seal global) và gửi về lưu trữ trên Bảng dữ liệu sự kiện tại Phương tiện lưu trữ tập tin.

- Mỗi khi nhận được thông tin sự kiện đối với tập tin từ tất cả các Phương tiện lưu trữ tập tin, hệ thống máy chủ sẽ tính con dấu khối toàn cầu và lưu trữ trên Bản dữ liệu sự kiện

trên hệ thống máy chủ kết nối Internet, cùng với sự kiện đó. Con dấu toàn cầu được tính theo công thức:

Block_Seal Global= SHA[block_Seal khối liền trước; thông tin sự kiện]

Dễ thấy, công thức tính con dấu khối toàn cầu cũng chính là công thức tính con dấu của khối cục bộ (Block_seal Local), tức là con dấu của khối thuộc tập tin được lưu trữ trên Phương tiện lưu trữ tập tin. Tuy nhiên, hai con dấu này là khác nhau. Lý do bởi vì chuỗi sự kiện của khối toàn cục là tất cả các sự kiện xảy ra với tất cả các tập tin có nguồn gốc từ tập tin khởi phát được lưu trữ tại Máy chủ; trong khi chuỗi sự kiện của khối cục bộ là sự kiện xảy ra với tập tin được lưu trữ trên Thiết bị lưu trữ tập tin mà thôi.

Ví dụ, khi có sự kiện (x) với tập tin được lưu trữ ở thiết bị A, là sự kiện thứ a tính từ sự kiện khởi phát đối với tập tin đó, sự kiện đó sẽ được lưu trữ cùng với con dấu khối cục bộ theo công thức: Block_Seal Local (A)= SHA[block_Seal (a-1); thông tin sự kiện (x)]

Tuy nhiên, cũng sự kiện đó được đồng bộ hóa lên Hệ thống máy chủ, nó trở thành sự kiện thứ b trong chuỗi sự kiện của tất cả các tập tin. Như vậy, con dấu khối toàn cầu sẽ được xác định theo công thức: Block_Seal Glocal (A)= SHA[block_Seal (b-1); thông tin sự kiện(x)]

Bằng cách ghi nhận đó, hệ thống biết được mọi sự kiện xảy ra với từng tập tin trên tất cả các Phương tiện lưu trữ tập tin, tạo điều kiện cho việc kiểm soát một cách thống nhất tất cả các tập tin trên tất cả các phương tiện lưu trữ, miễn là có kết nối internet.

Tập tin có entropy cao, sau khi đã được bảo vệ bằng phương pháp đề xuất tại sáng chế này được giải mã để sử dụng theo các bước sau đây:

Bước 1. thông tin tài khoản của người dùng, gồm mật khẩu, tên, sẽ được nạp để mở tập tin thùng chứa (container file).

Trên cơ sở đó, tham số LoginHash sẽ được tính toán theo công thức:

LoginHash = SHA(SecretKey, UserName, Password, P)

Trong đó, P là số thứ tự các lần thử kiểm tra quyền người dùng, được bắt đầu từ 1 đến [p]. Trong đó, P là quyền thứ p của người dùng.

Trong một phương án của sáng chế này, tham số p bằng 5, tương ứng với việc người dùng có 5 quyền là (1) quyền đọc tập tin; (2) quyền xem lịch sử sự kiện của tập tin; (3) quyền sao chép tập tin; (4) quyền ngừng sao chép tập tin và (5) quyền xóa tập tin. Tuy nhiên, số lượng quyền của người dùng không bị giới hạn bởi số quyền này.

Lần lượt tính các tham số LoginHash ứng với giá trị P từ 1 đến p, nếu tham số LoginHash tính được ứng với giá trị P là a (*a là số tự nhiên bất kỳ nằm trong khoảng từ 1 đến*

p) không trùng với Chữ ký số của mật khẩu (password_hash) của bất cứ người nào trong những người/nhóm người dùng đã được cấp phép đã lưu trên tập tin thùng chứa thì hệ thống sẽ tính lại tham số LoginHash với giá trị P bằng a+1, với điều kiện a+1 nhỏ hơn hoặc bằng p.

Nếu tham số LoginHash tính được ứng với tất cả các giá trị của P từ 1 đến p không trùng với Chữ ký số của mật khẩu (password_hash) của bất cứ người nào trong những người/nhóm người dùng đã được cấp phép đã lưu trên tập tin thùng chứa thì hệ thống sẽ coi là việc đăng nhập thất bại. Hệ thống tạo ra một khối (block) mới trên Bảng dữ liệu sự kiện của tập tin để ghi nhận sự kiện đăng nhập thất bại, đóng dấu (Block_Seal) lên sự kiện đó. Trong một phương án ưu tiên, sự kiện sẽ được cập nhật lên máy chủ có kết nối internet, đóng dấu toàn cầu (Block_Seal Global) lên sự kiện đó.

Nếu tham số LoginHash tính được ứng với giá trị P là a (*a là số tự nhiên bất kỳ nằm trong khoảng từ 1 đến p*) trùng với Chữ ký số của mật khẩu (password_hash) của người thứ k trong những người/nhóm người dùng đã được cấp phép đã lưu trên tập tin thùng chứa thì xác định đó là người dùng thứ k với quyền người dùng là a.

Bằng cách làm đó, người dùng không cần nhớ và nạp thông tin về quyền của mình vào hệ thống khi mở tập tin.

Bước 2. Giải mã thông tin người dùng thứ k bằng mật khẩu người dùng đã nhập theo công thức: UserInfo= AES.Decrypt(secret-key, password, ProtectedUserInfo)

Nếu tên người dùng (username) trong thông tin người dùng giải mã được trùng với tên người dùng đã nhập, tức là mật khẩu chuẩn xác.

Nếu không đúng, việc tính ra chữ ký số của mật khẩu của người dùng ở bước 1 chỉ là tình cờ (với xác xuất vô cùng nhỏ) và báo đăng nhập thất bại. Hệ thống tạo ra một khối (block) mới trên Bảng dữ liệu sự kiện của tập tin để ghi nhận sự kiện đăng nhập thất bại, đóng dấu (Block_Seal) lên sự kiện đó. Trong một phương án ưu tiên, sự kiện này sẽ được cập nhật lên máy chủ có kết nối internet, đóng dấu toàn cầu (Block_Seal Global) lên sự kiện đó.

Bước 3. Trên cơ sở mật khẩu người dùng, Khóa mã được bảo vệ (protectedCypherKey) được giải mã để lấy Khóa mã (CypherKey) theo công thức:

CypherKey =AES.Decrypt(secret-key, password, protectedCypherKey)

Bước 4. Sử dụng CypherKey để giải mã dữ liệu đã được bảo vệ bằng mã hoán vị các thành tố của bản mã.

Sáng chế cũng đề xuất hệ thống thực hiện phương pháp bảo vệ tập tin có entropy thông tin cao bằng cách sử dụng kết hợp mã hoán vị, chuẩn mã hóa EAS và công nghệ chuỗi khối. Hệ thống theo sáng chế này gồm 3 thành phần: (i) Thiết bị lưu trữ tập tin; (ii) thiết bị/phần

mềm có khả năng mã hóa/giải mã tập tin và (iii) máy chủ kết nối internet có khả năng lưu trữ thông tin sự kiện liên quan đến tập tin hoặc máy chủ kết nối internet có khả năng lưu trữ tập tin và toàn bộ các thông tin sự kiện liên quan đến tập tin và tất cả các bản sao của tập tin được lưu trữ trên các Thiết bị lưu trữ tập tin đơn lẻ. Chi tiết các thành phần này và cách hệ thống hoạt động đã được mô tả ở phần trên.

YÊU CẦU BẢO HỘ

1. Phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao , bao gồm:

bước 1: sinh khóa mã (CypherKey) một cách ngẫu nhiên, sau đó mã hóa tập tin bằng mã hoán vị khóa mã (CypherKey) để tao ra bản mã theo công thức sau:

$$DataFile_{bản\ mã} = Swap(CypherKey, DataFile_{nội\ dung})$$

trong đó $DataFile_{bản\ mã}$ là bản mã sau khi tập tin được mã hóa; $DataFile_{nội\ dung}$ là nội dung của tập tin khi chưa mã hóa, CypherKey là khóa mã, và Swap là hàm hoán vị;

bước 2: đặt mật khẩu để bảo vệ tập tin, hệ thống tính toán chữ ký số (hash) của mật khẩu của người dùng theo công thức:

password_Hash= SHA(secretkey, username, password, permission)

trong đó,

- secret-key là khóa bí mật, riêng có của hệ thống và được ẩn đi,
- username là tên người dùng,
- password là mật khẩu do người dùng thiết lập,
- permission là quyền của người dùng đối với tập tin,

bước 3: mã hóa khóa mã và thông tin tài khoản người dùng bằng chuẩn mã hóa đối xứng.

2. Phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao theo điểm 1, trong đó

chuẩn mã hóa đối xứng được chọn từ một trong số các chuẩn mã hóa đối xứng bao gồm Twofish, Serpent, AES, Blowfish, CAST5, RC4, Tam phần DES (Triple DES), và IDEA (International Data Encryption Algorithm).

3. Phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao theo điểm 1, trong đó chuẩn mã hóa đối xứng là AES, trong đó:

- thông tin tài khoản người dùng được mã hóa theo công thức:

ProtectedUserInfo = AES.Encrypt(secretkey, username, password, permission)

- mỗi người sẽ có một Khóa mã (CypherKey) được mã h`óa riêng theo công thức:

ProtectedCypherKey=AES.Encrypt(secretkey, password, CypherKey).

4. Phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao theo điểm 1, trong đó phương pháp còn bao gồm các bước sau :

bước 4: tạo ra tập tin "thùng chứa" (container file) để chứa toàn bộ các thông tin của tập tin đã được mã hóa sẽ gồm:

(i) bản mã ($datafile_{bản\ mã}$),

(ii) chữ ký số của mật khẩu của tất cả những người/ nhóm người dùng đã được cấp phép (password_hash),

(iii) thông tin đã được mã hóa về tất cả các người dùng/ nhóm người dùng (protected UserInfo) và tất cả các khóa mã (protectedCypherkey), và

bước 5: tạo ra bảng dữ liệu sự kiện, lưu trữ toàn bộ lịch sử của tập tin từ lúc được tạo ra (khởi phát) và các sự kiện xảy ra với đến tập tin đó sau thời điểm khởi phát theo công nghệ chuỗi khối, và lưu trữ bảng dữ liệu sự kiện này trên một phương tiện lưu trữ tập tin như là một phần mở rộng của tập tin thùng chứa hoặc như một tập tin riêng;

trong đó khối khởi phát và các khối tiếp theo trong chuỗi khối sẽ được đóng dấu theo công thức: Block_Seal= SHA[block_Seal khối liền trước; thông tin sự kiện]; trong đó, nếu khối đầu tiên là khối khởi phát thì tham số "block_seal khối liền trước" được đặt bằng giá trị nul;

nếu người dùng sao chép một tập tin thì một khối mới trên tập tin gốc sẽ được tạo ra để ghi nhận sự kiện này; đồng thời, trên tập tin mới được tạo ra thì sự kiện sao chép sẽ được coi là sự kiện khởi phát của tập tin mới được tạo ra đó.

5. Phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao theo điểm 4, trong đó, phương pháp còn bao gồm bước 6: đồng bộ hóa bảng dữ liệu sự kiện tại phương tiện lưu trữ tập tin tại bước 5 với bảng dữ liệu sự kiện trên hệ thống máy chủ kết nối internet ; theo đó,

- trên máy chủ có chứa Bảng dữ liệu sự kiện toàn cầu, chứa tất cả các thông tin liên quan đến tập tin, theo đó, thông tin sự kiện đối với tập tin gốc và tất cả các tập tin được sao chép từ tập tin gốc được lưu trữ trên tất cả các Phương tiện lưu trữ tập tin sẽ được cập nhật trên Bảng dữ liệu sự kiện toàn cầu thông qua hệ thống internet;

ứng với mỗi sự kiện sẽ có một khối mới được tạo ra, cùng với một con dấu khối toàn cầu; con dấu khối toàn cầu được tính theo công thức:

Block_Seal Global= SHA[block_Seal khối liền trước; thông tin sự kiện]

- trên Bảng dữ liệu sự kiện tại Phương tiện lưu trữ tập tin có thêm trường thông tin ghi nhận con dấu khối toàn cầu; theo đó, khi thông tin sự kiện được cập nhật với hệ thống toàn cầu, một con dấu khối toàn cầu được tạo ra và gửi về Phương tiện lưu trữ tập tin để lưu trữ.

6. Phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao theo điểm 5, khác biệt ở chỗ:

ngoài các thông tin sự kiện liên quan đến tập tin, tập tin gốc đồng thời được lưu trữ trên hệ thống máy chủ, cho phép người dùng thực hiện quyền của mình đối với tập tin một cách trực tiếp trên máy chủ, gồm tìm kiếm, truy cập để xem, tải về phương tiện lưu trữ, xóa hoặc cấp quyền cho người dùng và các quyền khác.

7. Phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao theo điểm 1, trong đó, việc mã hóa tập tin bằng khóa mã ở bước 1 được thực hiện bằng cách hoán vị các tiểu phân đoạn trong từng phân đoạn của tập tin dựa trên khóa mã lần lượt theo quy trình sau:

(i) phân chia tập tin thành nhiều phân đoạn bằng nhau, giá trị của mỗi phân đoạn là $2^n \times 256$ byte, trong đó n là số nguyên không âm, nếu việc phân chia có số dư thì phần dư sẽ được coi là một phân đoạn độc lập riêng và phần phân đoạn này không được mã hóa; và ;chia mỗi phân đoạn thành 256 tiểu phân đoạn bằng nhau, mỗi tiểu phân đoạn sẽ có giá trị tương ứng bằng 2^n byte;

(ii) mô tả trình tự sắp xếp các tiểu phân đoạn với độ dài 256 phần tử bằng một khóa mã (Cypherkey) được tạo ra một cách ngẫu nhiên; và

(iii) hoán vị trình tự tiểu phân đoạn trong tất cả các phân đoạn, từ trình tự thông thường (lần lượt từ 0 đến 255) sang trình tự do khóa mã quy định. .

8. Phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao theo điểm 7, trong đó, tham số n có giá trị từ 2 đến 4.

9. Phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao theo điểm 1, trong đó, việc mã hóa tập tin bằng khóa mã ở bước 1 được thực hiện bằng cách hoán vị từ mã như sau:

(i) mô tả trình tự sắp xếp các từ mã có số lượng phần tử bằng với số lượng từ mã bằng một khóa mã được tạo ra một cách ngẫu nhiên, và

(ii) hoán vị từ mã trên tập tin thành từ mã tương ứng do khóa mã quy định theo nguyên tắc hoán vị từng từ mã.

10. Phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao theo điểm 9, trong đó, số lượng từ mã là 256.

11. Phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao theo điểm 1, trong đó, phương pháp còn bao gồm bước 1b sau bước 1:

bước 1b: chia bản mã thành một hoặc nhiều khối, và gán một chữ ký số cho mỗi khối của bản mã , trong đó chữ ký số của từng khối được tính theo công thức:

Hash[j] = SHA(Hash[j-1], DataFile,DataBlock[j])

trong đó,

- SHA là hàm mã hóa 1 chiều,
- Hash[j-1] là chữ ký số của khối liền trước; nếu bản mã chỉ có duy nhất một khối hoặc là khối đầu tiên thì Hash[j-1] sẽ có giá trị nul,
- DataFile là tập tin sau khi đã được mã hóa (bản mã),
- DataBlock[j] là khối thứ j trong dữ liệu; nếu bản mã chỉ có duy nhất một khối hoặc là khối đầu tiên thì tham số DataBlock sẽ bằng với giá trị bản mã (DataFile).

12. Phương pháp bảo vệ nội dung tập tin có Entropy thông tin cao theo điểm 11, trong đó, mỗi khối được quy ước có dung lượng từ 1MB đến 16MB.

13. Hệ thống thực hiện phương pháp bảo vệ nội dung tập tin theo điểm 1, bao gồm:

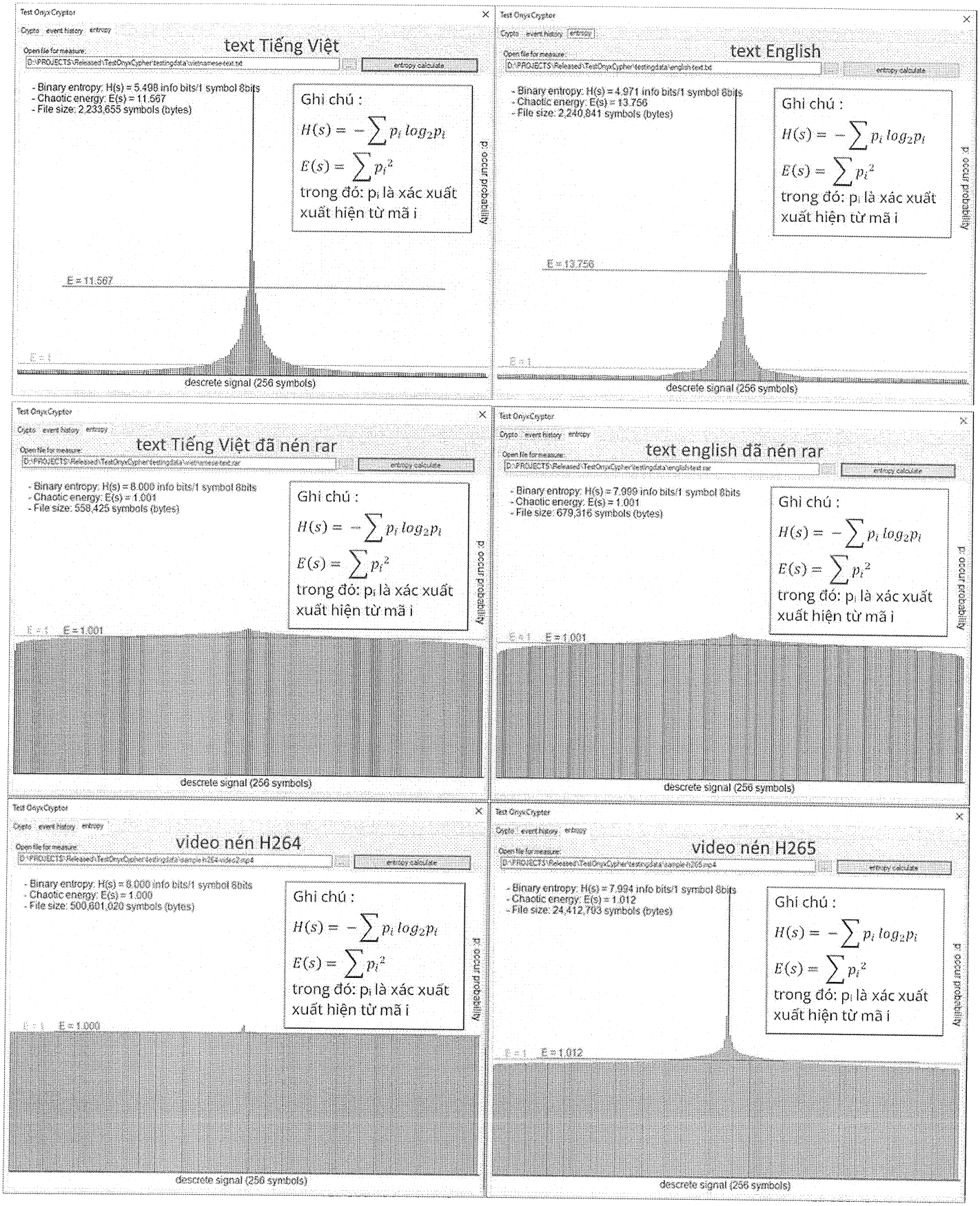
thiết bị lưu trữ và nén entropy cho tập tin; và

thiết bị mã hóa/giải mã tập tin dùng để mã hóa/giải mã tập tin đã nén entropy ở thiết bị lưu trữ và tính toán các tham số theo phương pháp theo điểm 1;

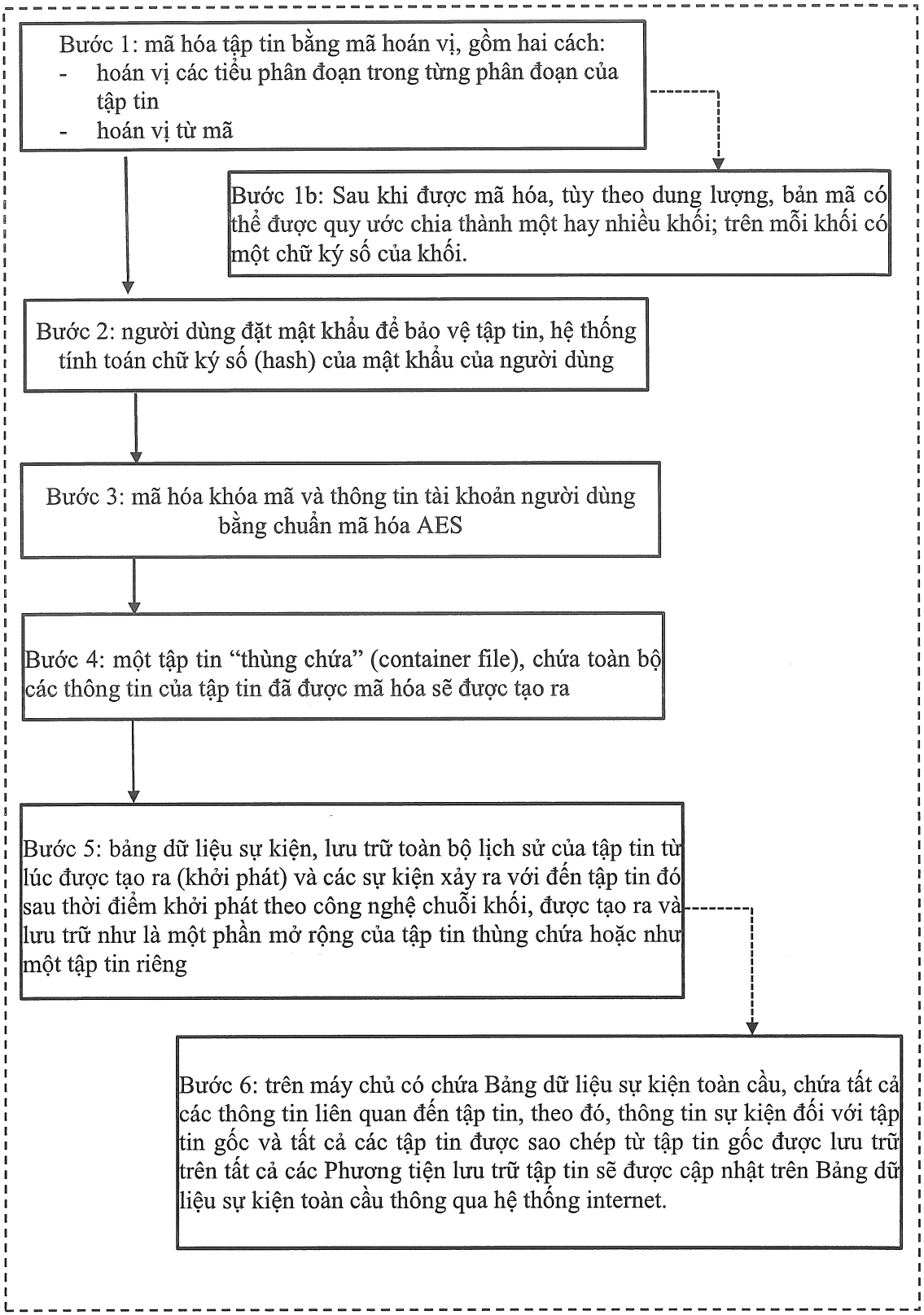
máy chủ có kết nối internet để lưu trữ thông tin về tập tin, từ sự kiện khởi phát đến các sự kiện khác có liên quan đến tập tin được lưu trữ trên thiết bị lưu trữ và nén entropi và được tạo ra ở thiết bị mã hóa/giải mã tập tin.

14. Hệ thống theo điểm 13, khác biệt ở chỗ, máy chủ có kết nối internet có khả năng lưu trữ thêm tập tin gốc.

Hình 1


text Tiếng Việt
- Binary entropy: H(s) = 5.498 info bits/1 symbol 8bits
- Chaotic energy: E(s) = 11.567
- File size: 2,233,655 symbols (bytes)
E = 11.567
descrete signal (256 symbols)
text English
- Binary entropy: H(s) = 4.971 info bits/1 symbol 8bits
- Chaotic energy: E(s) = 13.756
- File size: 2,240,841 symbols (bytes)
E = 13.756
descrete signal (256 symbols)
text Tiếng Việt đã nén rar
- Binary entropy: H(s) = 8.000 info bits/1 symbol 8bits
- Chaotic energy: E(s) = 1.001
- File size: 558,425 symbols (bytes)
E = 1.001
descrete signal (256 symbols)
text english đã nén rar
- Binary entropy: H(s) = 7.999 info bits/1 symbol 8bits
- Chaotic energy: E(s) = 1.001
- File size: 679,316 symbols (bytes)
E = 1.001
descrete signal (256 symbols)
video nén H264
- Binary entropy: H(s) = 8.000 info bits/1 symbol 8bits
- Chaotic energy: E(s) = 1.000
- File size: 500,601,020 symbols (bytes)
E = 1.000
descrete signal (256 symbols)
video nén H265
- Binary entropy: H(s) = 7.994 info bits/1 symbol 8bits
- Chaotic energy: E(s) = 1.012
- File size: 24,412,793 symbols (bytes)
E = 1.012
descrete signal (256 symbols)
Ghi chú :
$H(s) = -\sum p_i \log_2 p_i$
$E(s) = \sum p_i^2$
trong đó: p_i là xác xuất xuất hiện từ mã i
p: occur probability

Hình 2.


Bước 1: mã hóa tập tin bằng mã hoán vị, gồm hai cách:
- hoán vị các tiểu phân đoạn trong từng phân đoạn của tập tin
- hoán vị từ mã
Bước 1b: Sau khi được mã hóa, tùy theo dung lượng, bản mã có thể được quy ước chia thành một hay nhiều khối; trên mỗi khối có một chữ ký số của khối.
Bước 2: người dùng đặt mật khẩu để bảo vệ tập tin, hệ thống tính toán chữ ký số (hash) của mật khẩu của người dùng
Bước 3: mã hóa khóa mã và thông tin tài khoản người dùng bằng chuẩn mã hóa AES
Bước 4: một tập tin "thùng chứa" (container file), chứa toàn bộ các thông tin của tập tin đã được mã hóa sẽ được tạo ra
Bước 5: bảng dữ liệu sự kiện, lưu trữ toàn bộ lịch sử của tập tin từ lúc được tạo ra (khởi phát) và các sự kiện xảy ra với đến tập tin đó sau thời điểm khởi phát theo công nghệ chuỗi khối, được tạo ra và lưu trữ như là một phần mở rộng của tập tin thùng chứa hoặc như một tập tin riêng
Bước 6: trên máy chủ có chứa Bảng dữ liệu sự kiện toàn cầu, chứa tất cả các thông tin liên quan đến tập tin, theo đó, thông tin sự kiện đối với tập tin gốc và tất cả các tập tin được sao chép từ tập tin gốc được lưu trữ trên tất cả các Phương tiện lưu trữ tập tin sẽ được cập nhật trên Bảng dữ liệu sự kiện toàn cầu thông qua hệ thống internet.