Journal of educational equipment: Applied research, Volume 2, Issue 325 (November 2024)
ISSN 1859 - 0810

Nghiên cứu và đề xuất giải pháp để bảo vệ dữ liệu trong chuyển đổi số

Hồ Ngọc Vinh*, Lê Thị Thu Huyền**

**Trường Đại học Sư phạm Kỹ thuật Vinh*
*** Bệnh viên Ung bướu Nghệ An*
Received: 20/10/2024; Accepted: 26/10/24; Published: 6/11/2024

Abstract. *Network and Internet give a big benefit, it lets us ability to access and retreive information at anywhere. Beside these, araising many threat of illegal access to data. So the need of protecting information in the insecuring environment is very important. In this paper, we present a solution that stores data in secure virtual volume, which based on strong cryptography systems and deep knowledge of computer architecture.*

Keywords: *Security, data, programs, internets, virtual volume.*

1. Đặt vấn đề

Với sự phát triển của hạ tầng mạng và Internet, chúng ta có thể dễ dàng kết nối, chia sẻ cũng như tìm kiếm và truy cập đến thông tin từ bất kỳ đâu. Đi đôi với lợi ích đó là nhiều nguy cơ tiềm ẩn. Khi chưa kết nối, thông tin lưu trên máy tính khó có thể bị truy cập, đánh cắp nếu không trực tiếp sử dụng chính máy tính đó. Nhưng khi kết nối mạng, người ta có thể tấn công kiểm soát hay đánh cắp thông tin từ xa. Dẫn đến nhu cầu cần phải có những biện pháp, công cụ để bảo vệ thông tin trong môi trường không an toàn.

Có rất nhiều chương trình chống virus, tường lửa ngăn chặn truy cập bất hợp pháp và không ngừng được cập nhật. Đặc điểm chung của hầu hết các chương trình này là đi sau các chương trình nguy hiểm để bịt các lỗ hổng, chống lại các hình thức tấn công đã xuất hiện. Một cách khác là lưu trữ thông tin dưới dạng "ẩn", sử dụng các hệ mật mã. Với thông tin đã được mã hóa, ngay cả khi truy cập được vào hệ thống, kẻ đột nhập cũng không thể "đọc" được thông tin nếu không có được khóa giải mã thích hợp

Giải pháp đề xuất dựa trên cơ sở vững chắc của lý thuyết mã, ứng dụng các hệ mã đã được kiểm chứng kết hợp với các hiểu biết sâu về kiến trúc máy tính. Trong giải pháp này, chúng tôi xây dựng một không gian lưu trữ an toàn – phân vùng ảo, tại đó thông tin lưu được mã hóa và các truy cập đều được kiểm soát chặt chẽ.

2. Nội dung nghiên cứu

2.1. Nguyên lý bảo vệ dữ liệu

Muốn bảo vệ dữ liệu ta phải ngăn chặn được mọi hình thức tấn công nhằm biết được nội dung của dữ liệu cần bảo vệ. Thông thường dữ liệu có 2 trạng thái là:

+ *Trạng thái lưu trữ*: Là trạng thái của dữ liệu khi nội dung của chúng được lưu trữ trên các thiết bị lưu trữ.

+ *Trạng thái thao tác*: Là trạng thái của dữ liệu khi được các chương trình xử lý tương ứng nạp vào bộ nhớ để hiển thị, chỉnh sửa nội dung hay thực hiện một thao tác nào đó.

Do vậy, muốn bảo vệ dữ liệu một cách đầy đủ ta cần phải bảo vệ chúng ở cả hai trạng thái trên.

Hiện nay, các hệ điều hành như Microsoft Windows đều có những cơ chế nhằm bảo vệ các tài nguyên trên máy của người dùng (trong đó có các file dữ liệu) khỏi các cuộc tấn công đến từ bên ngoài. Cụ thể là áp dụng phương áp kiểm soát người dùng bằng tên và mật khẩu khi đăng nhập vào hệ thống, chỉ cho phép bên ngoài có thể truy cập được những tài nguyên đã được chia sẻ (sharing). Phương pháp này đã tạo ra một bức tường bảo vệ dữ liệu khá hiệu quả. Tuy nhiên nó sẽ trở thành vô dụng khi người dùng vô ý kích hoạt một chương trình gián điệp (spyware). Chương trình này có thể do người dùng vô ý download từ trên mạng hay một nguồn phần mềm không tin cậy có chứa spyware. Khi được kích hoạt, spyware sẽ có quyền của người dùng đã kích hoạt nó. Vì vậy nó hoàn toàn có thể truy cập vào hệ thống file, lấy đi những file dữ liệu quan trọng rồi chuyển ra ngoài mạng. Do đó để bảo vệ dữ liệu ở trạng thái lưu trữ, cần phải mã hóa chúng trước khi được lưu trữ trên ổ đĩa. Bài toán đặt ra là ta cần áp dụng thuật toán mã hoá và phương pháp sinh khoá như thế nào để bảo đảm độ an toàn của dữ liệu cần bảo vệ nhưng không làm ảnh hưởng nhiều đến tốc độ mã hóa và giải mã dữ liệu. Để đáp ứng hai yêu cầu trên, ta có thể sử dụng một số mã hoá đối xứng có độ mật cao hiện nay như AES với khoá có độ dài tương đối lớn (128 bit) được tạo từ các thông tin cá nhân của người sử dụng (chẳng hạn là tên và mật khẩu).

Đối với dữ liệu ở trạng thái thao tác, nội dung của chúng có thể được chứa trong bộ nhớ, hiện thị trên

màn hình hay được nhập vào từ bàn phím. Do vậy, muốn bảo vệ dữ liệu ở trạng thái này ta cần phải ngăn chặn được những hành động nhằm theo dõi và đọc nội dung từ các hệ thống tương ứng gồm bàn phím, màn hình và bộ nhớ.

**Bàn phím:* Những dữ liệu văn bản đều được nhập vào từ bàn phím, nếu theo dõi được quá trình gõ phím trên bàn phím, ta có thể đoán được phần lớn nội dung của văn bản đó. Mặt khác, các hệ điều hành đa nhiệm hiện nay đều có những cơ chế cho phép đưa vào hệ thống những chương trình xử lý bàn phím (keyboard hook procedure - KHP) thường với mục đích nhập các ký tự quốc tế không có sẵn trên bàn phím. Mã của phím trước khi đi tới chương trình soạn thảo được đưa qua các KHP đã đăng ký với hệ thống như sơ đồ trong Hình 1. Lợi dụng chính kẽ hở này, các chương trình theo dõi bàn phím (như KeyLog) đã đăng ký chúng với hệ thống để nhận được các kí tự đưa vào từ bàn phím. Các chương trình này ghi lại các ký tự nó nhận được ra các file log và chuyển chúng ra mạng để những người theo dõi có thể phân tích nội dung. Điều này đặc biệt nguy hiểm khi ta nhập những thông tin quan trọng như tên truy cập hay mật khẩu. Vì không thể ngăn cản được việc đăng ký các KHP với hệ thống nên để giải quyết triệt để vấn đề này, chúng tôi đề xuất phương án viết lại trình điều khiển keyboard. Thay vì trả cho hệ điều hành mã ASCII của ký tự nhận được, trình điều khiển keyboard mã hóa ký tự nhận được theo một thuật toán ngầm định giữa chương trình soạn thảo và trình điều khiển này. Khi nhận được các ký tự, chương trình soạn thảo sẽ tiến hành giải mã ngược để xác định ký tự thực được đưa vào.

Vì vậy, nếu trong hệ thống có những chương trình theo dõi bàn phím thì việc đọc nội dung đưa vào từ bàn phím cũng rất khó khăn vì chúng chỉ nhận được bản mã của các ký tự. Tuy nhiên, việc thực hiện phương pháp này cũng rất khó khăn vì phải đối mặt với vấn đề lập trình phần cứng. Hơn thế, có nhiều tiêu chuẩn cho các bàn phím, các giao diện với bàn phím rất đa dạng: có thể kết nối qua cổng PS2, USB, hồng ngoại (bàn phím không dây)… nên hầu như các trình điều khiển bàn phím đều do các hãng sản xuất phát triển. Nếu sử dụng phương pháp này, ta chỉ có thể thực hiện được với bàn phím chuẩn (có giao diện qua cổng PS2) và khuyến cáo người sử dụng dùng bàn phím loại này.

**Màn hình:* Khi dữ liệu được các chương trình hiển thị trên màn hình, nội dung của nó có thể bị lộ nếu người lấy cắp thông tin sử dụng chức năng chụp ảnh (capture) màn hình (tương tự như khi ta bấm nút Print Screen). Để bảo vệ màn hình, ta cần khống chế được chức năng này. Đối với hệ điều hành Windows, hiện không có một hàm nào của hệ thống cho phép khống chế việc chụp ảnh màn hình. Do vậy, để bảo vệ màn hình, chúng tôi đề xuất một phương án sử dụng kỹ thuật có tên "chồng hàm API". Đây là phương pháp sửa đổi lại con trỏ tới mã lệnh đầu tiên (entry) của một hàm được khai báo trong bảng ExportFunction của file thư viện động (dll) tương ứng của hệ thống để chúng trỏ tới hàm do ta tự định nghĩa. Trong trường hợp này ta sẽ thay đổi entry của hàm BitBlt trong thư viện GDI32.DLL.

**Bộ nhớ:* Thông thường, khi được nạp vào bộ nhớ để xử lý, dữ liệu sẽ được lưu dưới dạng bản rõ để tăng tốc độ thao tác. Nếu truy cập được vào vùng nhớ này thì dữ liệu sẽ dễ dàng bị đánh cắp. Do vậy, ta cần phải bảo vệ bộ nhớ chứa dữ liệu đang được thao tác. Tuy nhiên, các hệ điều hành đa nhiệm đã làm giúp ta nhiệm vụ này nhờ cơ chế quản lý bộ nhớ ảo. Mỗi tiến trình sẽ sử dụng một không gian nhớ riêng biệt không phụ thuộc vào nhau. Ánh xạ truy cập bộ nhớ của từng tiến trình chỉ là ánh xạ trên không gian nhớ của tiến trình đó. Mặt khác, việc cấp phát bộ nhớ cho từng tiến trình được hệ thống theo dõi rất chặt chẽ về quyền truy cập (đọc, ghi). Do vậy, nếu tiến trình truy cập vào vùng nhớ chưa được cấp phát hay không có đủ quyền truy cập thì sẽ bị hệ điều hành từ chối và báo lỗi truy cập (access violation). Vì vậy, dữ liệu lưu trong bộ nhớ sẽ chỉ được truy cập bởi chương trình xử lý dữ liệu tương ứng nên nó được bảo vệ an toàn.

**Nguyên lý bảo vệ chương trình:* Các loại dữ liệu thường có một hoặc một số chương trình tương ứng dùng để thao tác trên loại dữ liệu đó. Do vậy để bảo vệ dữ liệu, các chương trình này cần phải có cơ chế tự bảo vệ chính nó, chống lại sự tấn công của virus hay bị gỡ rối (debug) hoặc bẻ khóa (crack) để tìm hiểu các quá trình quan trọng như quá trình tạo khóa, xác thực người dùng, mã và giải mã dữ liệu…

Muốn chống lại những nguy cơ trên, chương trình sử dụng những phương pháp sau:

+ Sử dụng chữ ký số để tự kiểm tra: Khi mới được biên dịch, chương trình sẽ có một chữ ký gốc được xây dựng từ hàm băm. Chữ ký này sẽ được lưu vào một vùng bí mật của chương trình. Trong quá trình chạy, chương trình sẽ tự sinh lại chữ ký của chính nó rồi so sánh với chữ ký gốc trên. Nếu chúng khớp nhau có nghĩa là mã của chương trình không bị thay đổi do vậy có thể tiếp tục thực hiện. Nếu không khớp, tức là đã có một sự thay đổi bất thường trong code của chương trình. Sự thay đổi này chủ yếu do 2 nguyên nhân, bị nhiễm virus hoặc bị sửa mã nhằm mục đích nào đó. Trong những trường hợp này, chương trình sẽ tự động kết thúc nhằm trách những hậu quả xấu đối với dữ liệu mà nó xử lý.

+ Sử dụng các hàm inline để mã hóa: Khác với các

hàm bình thường được trình biên dịch chuyển thành mã máy, các hàm inline không được biên dịch như vậy. Thay vào đó, mỗi khi có lời gọi đến một hàm inline, trình biên dịch sẽ chèn toàn bộ đoạn mã của hàm inline tương ứng vào vị trí gọi hàm. Vì vậy, hàm inline sẽ làm cho kích thước của chương trình lớn hơn nhưng sẽ chạy nhanh hơn và đặc biệt là sẽ làm rối chương trình (ở dạng mã máy) ở những điểm gọi hàm inline. Đây chính là đặc điểm nổi bật của hàm inline trong việc chống bẻ khóa chương trình.

+ *Sử dụng kỹ thuật "bom thời gian" và hàm băm một chiều khi so sánh các thông tin quan trọng nhằm trách bị theo dõi*: Những điểm vào ra quan trọng của chương trình đều được cài những đoạn mã theo dõi thời gian thực hiện. Nếu phát hiện thời gian thực hiện quá lâu thì rất có thể nó đang bị debug và sẽ làm "bom nổ" tức là chương trình bị kết thúc tức thì. Để chống lại việc bị bẻ khóa chương trình bằng cách bỏ qua các lệnh kiểm tra quan trọng, ta cần sử dụng các hàm băm một chiều. Thay vì kiểm tra các dữ liệu quan trọng một cách trực tiếp, ta lấy hàm băm của chúng rồi so sánh các kết quả này với nhau. Phương pháp này kết hợp với việc sử dụng hàm inline đối với các hàm băm sẽ làm rối những đoạn lệnh cần bảo vệ. Vì vậy, nó khắc phục được những điểm yếu ở những đoạn so sánh bình thường.

2.2. Một hướng tiếp cận xây dựng hệ thống bảo vệ dữ liệu

Trên cơ sở những nguyên lý bảo vệ dữ liệu và chương trình đã phân tích ở phần trên, chúng tôi đề xuất một hướng tiếp cận trong việc xây dựng hệ thống bảo vệ dữ liệu trên môi trường không an toàn như sau:

+ *Các dữ liệu cần bảo vệ sẽ được lưu trữ trên "phân vùng ảo"*: Đây là một cấu trúc đặc biệt mà trên hệ điều hành chủ chỉ nhìn thấy dưới dạng một file hay một phân vùng của đĩa cứng nhưng hoàn toàn không hiểu cấu trúc bên trong nó. Trên phân vùng ảo, dữ liệu sẽ được lưu trữ dưới dạng mã sử dụng những thuật toán mã hóa có độ mật cao với khóa được tạo ra từ các thông tin người dùng.

+ Để truy xuất dữ liệu trên cấu trúc này ta cần xây dựng một trình điều khiển vào ra trên *phân vùng ảo*: Trình điều khiển này có nhiệm vụ cung cấp những hàm vào ra cơ bản qua đó xây dựng một cơ chế kiểm soát chặt chẽ và an toàn trong việc truy xuất dữ liệu trên phân vùng ảo. Đồng thời trình điều khiển cũng được áp dụng những phương pháp bảo vệ chương trình nhằm tránh những nguy cơ có thể xảy ra với nó.

Xây dựng các công cụ an toàn nhằm trợ giúp người dùng làm việc thuận tiện hơn trên môi trường phân vùng ảo.

2.2.1. Cấu trúc của phân vùng ảo

Như phân tích ở phần 2, để bảo vệ dữ liệu ta chỉ cần mã hóa chúng trước khi lưu trữ. Tuy nhiên, nếu mã hóa từng file như vậy thì rất khó khăn và bất tiện trong việc quản lý. Hơn thế, muốn thao tác với dữ liệu này ta cần phải giải mã dữ liệu trước khi tiến hành các thao tác xử lý hay xem nội dung. Đây chính là những thời điểm không an toàn với dữ liệu bởi nó được lưu trữ ở dạng rõ. Để giải quyết những nhược điểm trên thì phân vùng ảo là một giải pháp. Khi các dữ liệu quan trọng được tập trung lưu trữ trên phân vùng ảo, ta có thể áp dụng những phương pháp bảo vệ dữ liệu một cách thống nhất. Mặt khác, cách lưu trữ này cũng làm tăng độ an toàn của dữ liệu bởi việc lấy được dữ liệu từ file trên phân vùng ảo một cách trực tiếp cũng rất khó khăn vì sự phức tạp trong cấu trúc của phân vùng ảo.

Những thông tin cấu hình chung sẽ được lưu trữ ở phần đầu của phân vùng ảo. Phần không gian lưu trữ còn lại được chia thành các block có kích thước như nhau. Các block liên tiếp nhau được nhóm lại thành các phân vùng. Mỗi phân vùng có số lượng block như nhau và trạng thái của các block này được quản lý bởi bảng thông tin trạng thái block nằm ở đầu mỗi phân vùng.

**Bảng thông tin phân vùng:* Phân vùng ảo bắt đầu bằng bảng thông tin phân vùng. Giống như boot sector trong mô hình FAT. Bảng này chứa toàn bộ thông tin quy định khuôn dạng của chính phân vùng đó gồm tên, thuộc tính, phiên bản, serial, kích thước các block (byte) và kích thước phân vùng (block) của phân vùng ảo…. Do dữ liệu trên phân vùng ảo được mã hóa với khóa được sinh từ những thông tin người dùng. Vì vậy, khi mở phân vùng ta cần kiểm tra xem người dùng đã đăng nhập đúng hay sai. Mặt khác để đảm bảo an toàn cho dữ liệu trên phân vùng ảo, ta không được lưu bất kỳ thông tin nào trên phân vùng mà từ đó có thể khôi phục được khóa. Để giải quyết những yêu cầu này, bảng thông tin phân vùng có chứa các trường là kết quả hàm băn một chiều của các thông tin cần sử dụng khi xác thực người dùng.

Ngoài ra, nó con chứa một số thông tin nhằm bảo đảm an toàn cho phân vùng ảo như trường số lượng bản sao và chữ ký số. Các bảng hệ thống của phân vùng ảo sẽ được sao lưu với số lượng được chỉ ra trong trường số lượng bản sao. Khi có sự cố xảy ra, có thể do máy bị tắt đột ngột khi đang thao tác dữ liệu trên phân vùng ảo hoặc sector trong khu vực lưu trữ bảng thông tin này bị hỏng. Những sự cố này làm cho chữ ký gốc của bảng thông tin phân vùng không còn khớp với chữ ký của bảng thông tin phân vùng hiện thời. Trong trường hợp này, các bảng lưu dự phòng sẽ được huy động để sửa lỗi cho phân vùng.

**Phân vùng dữ liệu:* Theo sau bảng thông tin phân

vùng là các phân vùng dữ liệu. Việc nhóm các block thành phân vùng nhằm tạo cho phân vùng có khả năng mở rộng về kích thước. Nếu không có các phân vùng thì sẽ chỉ có một bảng thông tin trạng thái duy nhất. Những kích thước của bảng này là cố định nên ta không thể mở rộng phân vùng được. Nếu tách chúng thành từng phân vùng thì việc mở rộng phân vùng có thể được thực hiện bằng cách thêm các phân vùng mới vào cuối file chứa phân vùng. Mỗi phân vùng chỉ quản lý các block thuộc phân vùng đó thông qua bảng thông tin trạng thái block. Bảng này gồm một danh sách các phần tử, mỗi phần tử lưu trữ trạng thái của block tương ứng với nó. Khi trình quản lý phân vùng ảo hoạt động nó sẽ tổng hợp tất cả các bảng thông tin block trên các phân vùng để tạo thành một bảng thông tin block chung biến phân vùng trở thành một cấu trúc giống như hệ thống FAT32.

** Lưu trữ file và thư mục trên phân vùng ảo:* Các file được lưu trữ trên phân vùng ảo dưới dạng một tập các block có thứ tự xác định. Thứ tự này được chỉ ra bởi các bảng thông tin block ở dạng danh sách móc nối. Cấu trúc cây thư mục trong phân vùng được lưu trữ theo mô hình đệ quy, bắt đầu từ thư mục gốc có block đầu tiên được chỉ ra tại bảng thông tin phân vùng. Nội dung của một thư mục là một danh sách các điểm vào (entry) định danh các file và thư mục con tương ứng nằm trong thư mục đó. Mỗi entry là một bảng thông tin mô tả file

2.2.2. Trình quản lý vào ra trên phân vùng ảo

Chính vì có cấu trúc phức tạp, việc truy xuất dữ liệu trên phân vùng ảo phải thông qua một trình quản lý. Do đó ta có thể thiết lập những cơ chế kiểm soát chặt chẽ việc vào ra trên phân vùng ảo. Cụ thể, hệ thống bảo vệ dữ liệu đưa ra một mô hình vào ra được kiểm soát bởi một thẻ đặc biệt. Mỗi chương trình khi đăng nhập thành công với trình quản lý phân vùng ảo sẽ được cấp cho một thẻ đặc biệt. Bất kỳ một thao tác dữ liệu nào cũng đòi hỏi chương trình gọi phải xuất trình thẻ trên. Nếu thẻ này không hợp lệ thì thao tác sẽ bị từ chối. Do vậy, nếu bị truy cập bất hợp phát từ mạng thì chương trình đó cũng không thể có được thẻ trên vì vậy, mọi thao tác trên phân vùng đều bị từ chối.

Việc truyền tên và mật khẩu khi đăng nhập phân vùng được bảo vệ bằng mã khóa công khai (RSA). Do vậy tránh được việc lấy cắp những thông tin người dùng trong quá trình đăng nhập.

2.2.3. Xây dựng các tiện ích trên phân vùng ảo

Để trợ giúp người dùng trong quá trình sử dụng phân vùng ảo, ta cần xây dựng một số công cụ an toàn phục vụ người dùng. Khi trình bày bài viết này, chúng tôi đã xây dựng thành công một số tiện ích quan trọng sau:

** Tiện ích tìm kiếm mờ trên phân vùng ảo:* Đây là một công cụ đặc biệt để phục vụ cho việc tìm kiếm tài liệu trên phân vùng ảo. Nó ứng dụng thuật toán tìm kiếm mờ để tìm kiếm thông tin trong các file chứa tài liệu lưu trên phân vùng ảo hoặc ổ đĩa thực của hệ điều hành. Thuật toán tìm kiếm này dựa vào 2 tham số để đánh hạng cho kết quả tìm kiếm. Đó là tần xuất xuất hiện các từ khóa trong xâu mẫu và độ bảo toàn thứ tự của các từ khóa này trong những văn bản cần tìm. Việc kiểm tra sự xuất hiện của các từ khóa trong văn bản cần tìm được hiểu theo nghĩa mờ tức là: Với một ngưỡng mờ x% cho trước, nếu từ khóa A_0 của xâu mẫu giống với từ A_1 của văn bản cần tìm x% thì coi A_0 trùng với A_1. Để đánh giá độ mờ giữa 2 từ, tiện ích sử dụng thuật toán LCS làm cơ sở cho việc so sánh và cải tiến một số điểm trong thuật toán này để nó có thể đánh giá độ mờ của các từ tiếng Việt. Việc đánh giá độ bảo toàn thứ tự của các từ trong câu được thực hiện bằng thuật toán đếm số thuận thể.

** Tiện ích thao tác file trên phân vùng ảo:* Sử dụng những hàm vào ra cơ bản của trình quản lý phân vùng ảo, tiện ích này có thể trợ giúp người dùng dễ dàng tạo ra các file và thư mục trên phân vùng ảo bằng cách tạo mới hay sao chép, di chuyển các file hoặc cả cây thư mục từ phân vùng ảo này đến phân vùng ảo khác hay từ ổ đĩa của hệ điều hành vào phân vùng ảo hoặc ngược lại.

3. Kết luận

Xuất phát từ tình hình thực tế về sự thiếu hụt những công cụ bảo vệ dữ liệu trên môi trường không an toàn. Kết hợp với những cơ sở lý thuyết về mã hóa và chữ ký số, chúng tôi đã xây dựng được một hệ thống bảo vệ dữ liệu trên môi trường không an toàn và một số tiện ích quan trọng và an toàn trợ giúp người dùng trên môi trường này.

Khi hoàn chỉnh, chương trình sẽ mang lại hiệu quả lớn trong việc bảo vệ dữ liệu quan trọng trên môi trường không an toàn, bên cạnh các tiện ích khác tăng cường sự bảo vệ dữ liệu trọng yếu cho người dùng và môi trường đa dạng hiện nay.

Tài liệu tham khảo

[1] Phan Trung Huy (2012), "*Ứng dụng mã hoá và chữ ký số trong lĩnh vực bảo vệ bản quyền phần mềm*", Đề tài cấp Bộ 2012.

[2] D. R. Stinson (2019), *Cryptography in Theory and Practice*, CRC Press Taylor & Francis Group, FL 33487-2742.

[3] Ronald Rivest (20224), *The MD5 Message Digest Algorithm*, Jul 29, 2024

[4] Xiaoyun Wang, Dengguo Feng, Xuejia Lai, Hongbo Yu (2024), *Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD,* rump session, CRYPTO 2024.

[5] Announcing the Advanced Encryption