# Sử dụng mạng máy tính phân tán trong xây dựng giải pháp thám mã khối

**Hoàng Thái Hổ, Nguyễn Thế Hùng, Nguyễn Tuấn Minh**

***Tóm tắt*—Bài báo trình bày một giải pháp sử dụng năng lực của mạng máy tính phân tán cho thám mã khối. Hệ thống có cấu trúc dựa trên 3 phần mềm. Phần mềm quản trị sử dụng cho nhập dữ liệu đầu vào, phân tích và chia khoảng không gian khóa và phân tích kết quả. Phần mềm thám mã trên CPU và GPU được cài đặt tương ứng cho các máy tính trong mạng phân tán có nhiệm vụ thám mã đối với dữ liệu phần mềm quản trị cung cấp. Kết quả được gửi về phần mềm quản trị để phân tích và giải mã. Quá trình thám mã được thực hiện cùng lúc trên toàn bộ máy tính trong mạng vào thời gian máy tính nhàn rỗi, không ảnh hưởng tới hoạt động hàng ngày của người dùng. Hệ thống bao gồm cả các máy tính có sử card GPU giúp tăng hiệu suất thám mã lên gấp 11 lần. Giải pháp đã được ứng dụng trong thám mật khẩu Windows qua mã băm LAN Manager.**

***Abstract*—This paper presents a method to use the capabilities of distributed computer networks in cryptanalysis of block ciphers. The system is structured based on 3 software. Management software for input data entry, analysis, and keyspace division. Cryptanalysis software on CPU and GPU is installed respectively for client computers in the distributed network is responsible for cryptanalysis of data provided by the management software. The results are sent to the administrative software for analysis and decoding. The encryption process is performed on all computers in the network at the same time in their spare time, without affecting the user's daily activities. The system includes GPU computers that increase the performance of the cryptanalysis by 11 times. This solution has been applied in Windows password detection via LAN Manager hash code.**

***Từ khóa—Thám mã, vét cạn, mã khối, mạng máy tính phân tán, DES.***

***Keywords—Cryptanalysis, brute-force, block cipher, distributed computer networks, DES.***

## I. TỔNG QUAN

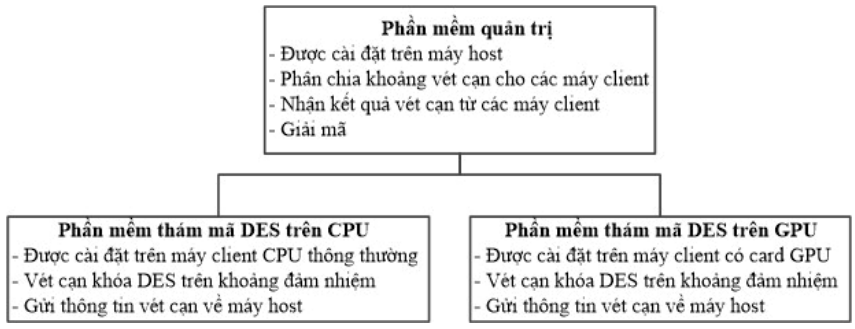
Công nghệ thông tin và viễn thông ngày càng phát triển, đòi hỏi nhu cầu về xử lý thông tin ngày càng cao, nhanh hơn nhiều lần tốc độ phát triển của tài nguyên phần cứng và phần mềm. Có nhiều vấn đề đặt ra mà các hệ thống tập trung thông thường không đáp ứng được, do tốc độ xử lý còn hạn chế [1]. Xây dựng một hệ thống phân tán có khả năng xử lý đồng thời một bài toán trên nhiều máy tính là một hướng giải quyết khả thi và đã được chứng minh tính hữu dụng. Hệ thống phân tán còn tạo nhiều thuận lợi trong việc chia sẻ thông tin trên toàn thế giới.

Với những ưu điểm nổi bật đó, hệ thống phân tán đã được ứng dụng vào nhiều lĩnh vực khác nhau như: mạng viễn thông (mạng điện thoại, điện thoại di động, mạng máy tính, mạng cảm biến không dây); các ứng dụng mạng (World Wide Web, các mạng ngang hàng, các trò chơi trực tuyến có nhiều người chơi, thực tế ảo, cơ sở dữ liệu phân tán và hệ quản trị...); kiểm soát quy trình thời gian thực các hệ thống điều khiển (điều khiển công nghiệp, điều khiển thiết bị bay,...); tính toán song song (điện toán đám mây, điện toán lưới,...) [2].

Trong mật mã học, việc ứng dụng hệ thống phân tán để tấn công vét cạn vào một số loại mã có thể nói không phải là điều mới. Điển hình như các kế hoạch tấn công của Distributed.Net lên các hệ mã: DES – các dự án DES I, II, III, RC5 – các dự án RC5-56, RC5-64 gần đây nhất là RC5-72 được cập nhật mới nhất vào tháng 02/2021 với hơn 142.000 máy tham gia. Việc thám mã sử dụng hệ thống máy tính phân tán sẽ tận dụng được tối đa phần cứng và thời gian rảnh rỗi của các máy tính trong mạng. Tuy nhiên đối với mỗi loại mã cần có những thuật toán thám mã cũng như cách thức phân chia công việc phù hợp để đạt được kết quả tối ưu nhất [3]-[7].

Dưới đây là giải pháp thám mã khối lấy đại diện DES bằng việc sử dụng phần cứng rảnh rỗi

Bài báo được nhận ngày 04/6/2021. Bài báo được nhận xét bởi phản biện thứ nhất ngày 10/9/2021 và được chấp nhận đăng ngày 10/10/2021. Bài báo được nhận xét bởi phản biện thứ hai ngày 18/10/2021 và được chấp nhận đăng ngày 20/10/2021.




Hình 1. Các phần mềm trong bộ công cụ thám mã.

trong hệ thống mạng máy tính phân tán. Việc phân phối công việc một cách tương đối đồng đều và ứng dụng xử lý song song đối với các máy tính có card GPU giúp tăng tối đa hiệu suất làm việc của các máy tính trong mạng.

## II. GIẢI PHÁP THÁM MÃ TRÊN MẠNG MÁY TÍNH PHÂN TÁN

Hệ thống mạng máy tính phân tán phục vụ cho thám mã DES sử dụng bộ công cụ thám mã gồm 3 phần mềm với các chức năng cơ bản được mô tả trong Hình 1.

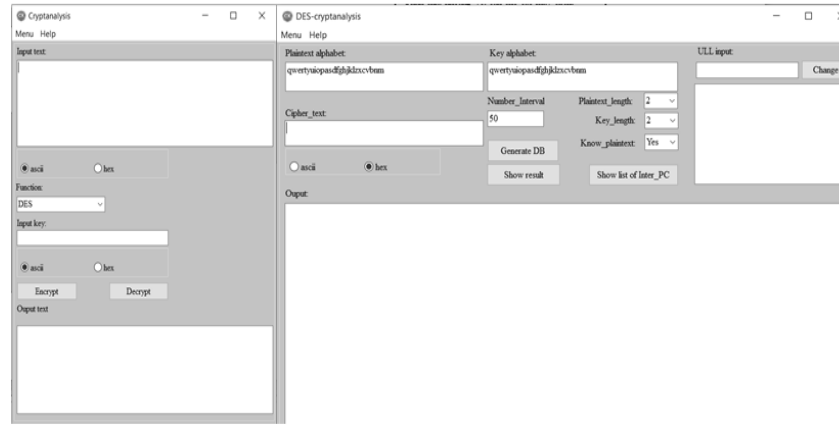
Phần mềm quản trị là phần mềm có giao diện được viết bằng ngôn ngữ $C^{++}$ trên môi trường $C^{++}$ Builder 10.3. Giao diện phần mềm được thể hiện trên Hình 2.

Phần mềm được cài đặt trên một máy quản trị trong mạng, có chức năng chính là chia khoảng vét cạn, quản lý quá trình giải mã và giải mã khi các máy trong mạng gửi kết quả về. Cụ thể:

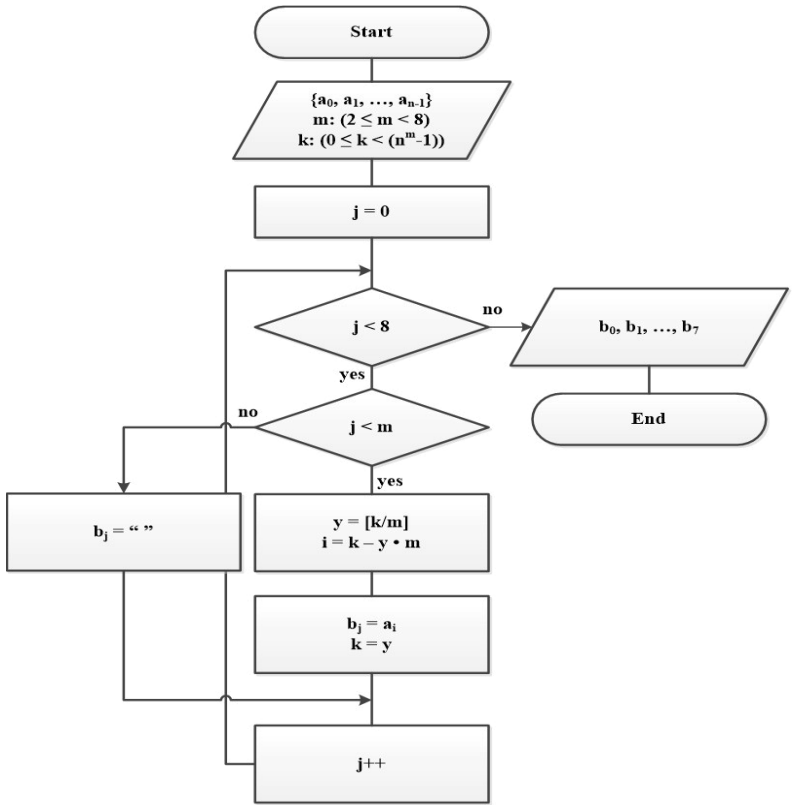
- Nhận dữ liệu đầu vào: Dữ liệu đầu vào bao gồm số lượng các khoảng cần chia: số lượng này không phụ thuộc vào số lượng máy tính có trong mạng; không gian các ký tự của khóa và của bản rõ: cho phép toàn bộ các ký tự trong hệ ASII và các ký tự nhập được từ bàn phím đối với không gian của khóa; độ dài của khóa và của bản rõ: cho phép tối đa là một khối 64 bit; bản mã: một khối bản mã 64 bit, máy host sẽ không gửi toàn bộ bản mã xuống các máy client cho quá trình thử khóa;

- Xử lý dữ liệu đầu vào: lọc các ký tự trùng nhau: để đảm bảo các ký tự trong không gian của bản rõ và khóa đều không bị lặp lại; lọc các ký tự có 7 bit đầu giống nhau đối với không gian các ký tự khóa và sắp xếp chúng theo thứ tự trong bảng ASCII 128: Trong mã hóa DES, khóa đầu vào là một khối 64 bit, tuy nhiên quá trình xử lý đầu vào các bit thứ 8 sẽ được loại bỏ thành khóa 56 bit do đó để giảm không gian các ký tự khóa phần mềm sẽ tự động lọc bỏ các ký tự có 7 bit đầu giống nhau. Ví dụ nếu trong không gian các ký tự khóa đầu vào chứa các ký tự *b* (**0100000**1), *c* (**0100000**1), *d* (**0100001**0), *e* (**0100001**1) thì phần mềm sẽ chỉ giữ lại *b* và *d* vì *b* và *c* có 7 bit đầu giống nhau tương tự đối với *d* và *e*. Như vậy không gian ký tự khóa sẽ được giảm xuống. Chẳng hạn đối với không gian đầu vào là "*abcdefghijklmnopqrstuvwxyz*" gồm 26 ký tự sẽ được giảm xuống còn 14 ký tự là "*abdfhjlnprtvxz*";

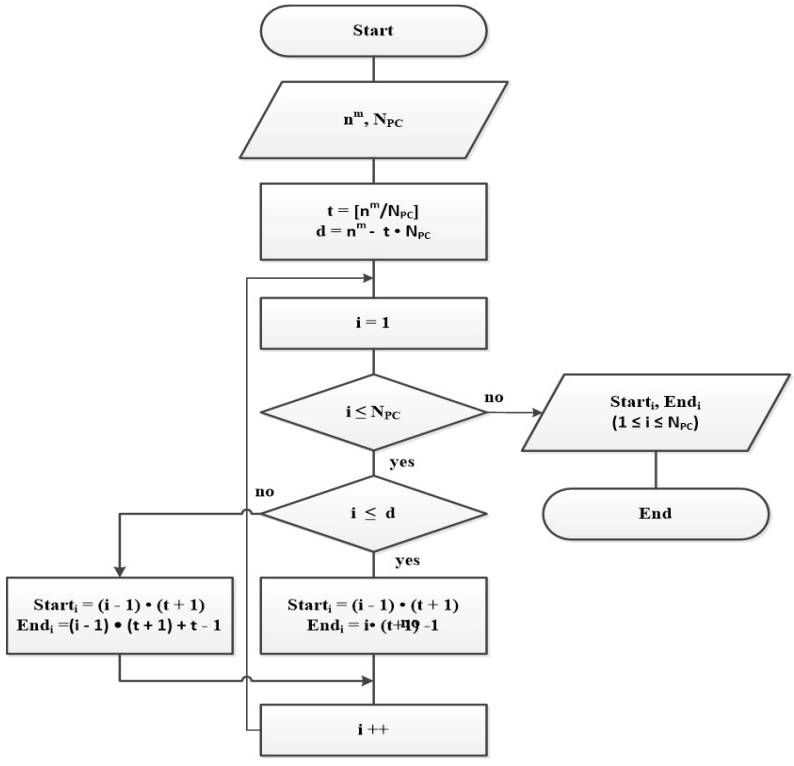
- Tính tổng số khóa dùng cho vét cạn: Giả sử không gian các ký tự khóa gồm $n$ ký tự $\{a_0, a_1, ..., a_{n-1}\}$, độ dài của khóa cần tạo là $m$ vì khóa đầu vào là một khối 64 bit, do đó $2 \le m < 8$ (trường hợp $m \# 7$, các ký tự còn lại của chuỗi sẽ được lấp đầy bằng ký tự trắng cho đủ 64 bit). Tổng số khóa có thể thành lập chính bằng $n^m$. Nếu gọi $k$ là số thứ tự của chuỗi tức là $0 \le k < n^m - 1$, thì với mỗi




Hình 2. Giao diện phần mềm quản trị.




Hình 3. Sơ đồ thuật toán tạo một chuỗi (8 ký tự) từ không gian các ký tự cho trước và số thứ tự của chuỗi.



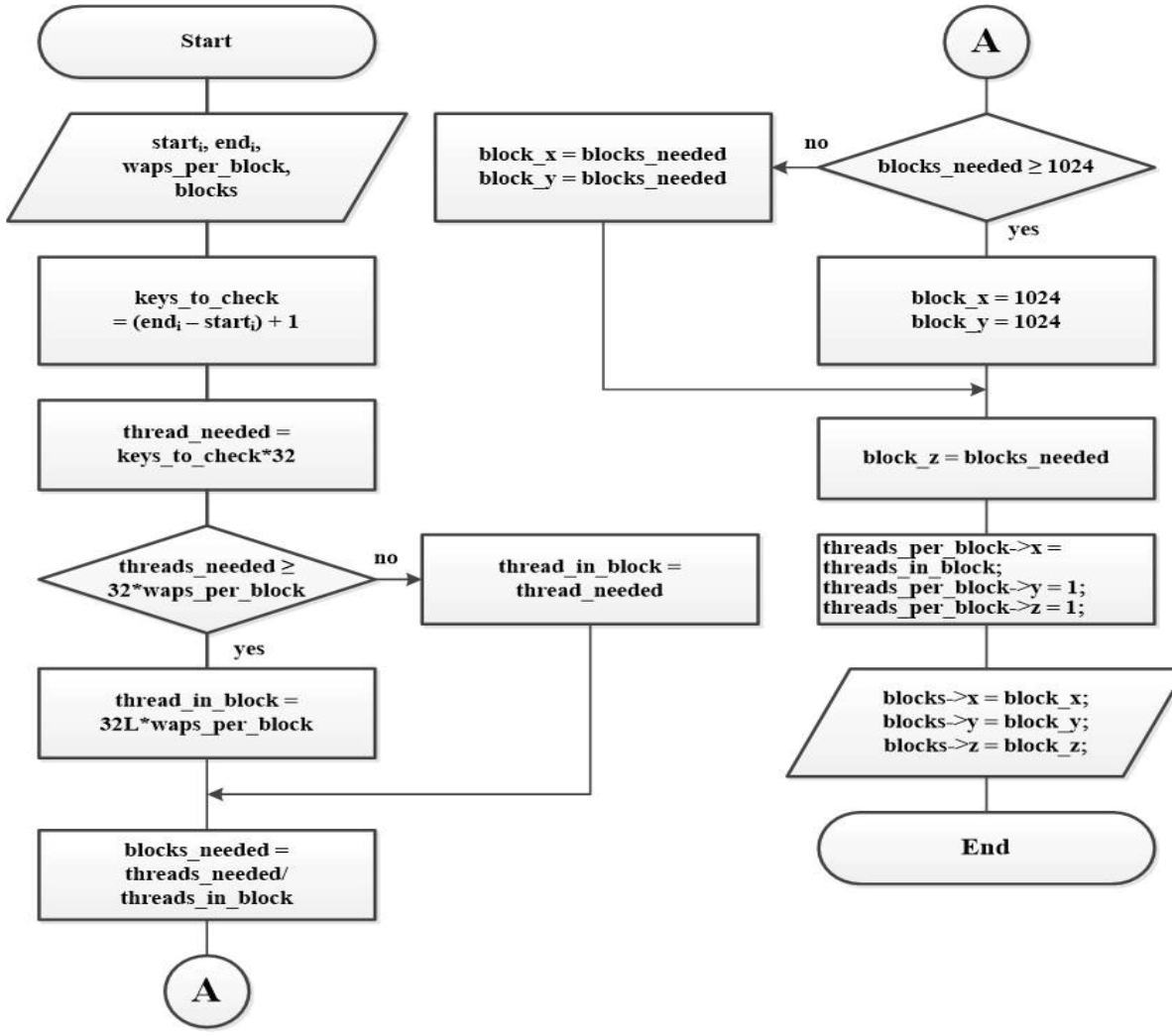

Hình 4. Sơ đồ thuật toán chia khoảng cho các máy tính trong mạng.

$k$ sẽ tạo được một chuỗi 64 bit riêng ($b_0b_1b_2b_3b_4b_5b_6b_7$). Thuật toán tạo chuỗi (khối 64 bit) ký tự khi biết trước độ dài và không gian các ký tự được trình bày trong Hình 3.

- Chia tổng số khoá cần thực hiện vét cạn thành từng khoảng và lưu vào file: Để chia không gian khóa cho các máy tính tham gia vét cạn một cách gần đều ta sử dụng sơ đồ thuật toán Hình 4. Ở đây ký hiệu $n^m$ được sử dụng lại là tổng số khóa cần thử, $N_{PC}$ là số lượng máy tính tham gia trong mạng. $Start_i$ và $End_i$ là ký hiệu số thứ tự bắt đầu và kết thúc của khóa cần thử trên máy tính thứ $i$. Theo đó nếu lấy tổng số khóa $n^m$ chia cho tổng số máy tính được giá trị là $t$ và dư $d$, thì từ máy tính thứ nhất tới máy tính thứ $d$ sẽ thực hiện thử lần lượt $(t + 1)$ khóa, các máy từ máy thứ $(d + 1)$ trở đi tới máy cuối cùng sẽ thử $t$ khóa. Do đó số lượng khóa được phân bổ tương đối đều cho các máy.

- Giải mã: Để đảm bảo tính an toàn, bảo mật đối với dữ liệu, quá trình giải mã toàn bộ bản mã sẽ được thực hiện trên phần mềm quản trị sau khi thu được khóa đúng từ kết quả thử khóa ở các máy client trong mạng.

Phần mềm thám mã DES trên CPU và GPU là phần mềm không có giao diện, được xây dựng bằng ngôn ngữ $C^{++}$ trên môi trường Visual Studio 2019. Các phần mềm này được cài đặt




Hình 5. Sơ đồ thuật toán xác định số lượng *block*s và *threads* trong card GPU.

lên các máy tính tham gia vào mạng theo độ tương thích có chức năng thử khóa và bản rõ trong mỗi khoảng đảm nhiệm. Cụ thể:

- Nhận dữ liệu đầu vào từ máy quản trị: Dữ liệu đầu vào cho phần mềm thám mã tại các máy trong mạng bao gồm: khối bản mã; không gian các ký tự của khóa và bản rõ; độ dài của khóa và bản rõ; khoảng vét cạn: mỗi máy tính sẽ chỉ thực hiện thử khóa trên một khoảng nhất định. Sau khi thực hiện xong nếu vẫn còn những khoảng chưa được xử lý, máy đó sẽ tự động lấy và xử lý tiếp, cứ như vậy và không theo thứ tự máy nào trước máy nào sau;
- Tính tổng số bản rõ dùng cho vét cạn;
- Tạo khóa trong khoảng vét cạn: Lần lượt tạo các trường hợp bản rõ có thể và mã hóa bằng thuật toán DES với khóa được tạo từ khoảng vét cạn;
- So sánh bản mã thu được với bản mã đầu vào và đưa ra kết quả;
- Gửi kết quả về máy quản trị sau khi thực hiện xong mỗi khoảng.

Đối với phần mềm chạy trên máy tính có card GPU được ứng dụng xử lý song song trên các tiểu trình (*threads*) để tăng tốc quá trình thử khóa. Việc tính toán số lượng khối (*blocks*) cần thiết theo các chiều (*x*, *y*, *z*) và các *threads* cần thiết trong mỗi *block* trên cơ sở các khoảng phân phối khóa cho trước được mô tả trong thuật toán Hình 5.

Ở đây chúng ta sử dụng mô hình *3D grid of 1d blocks* tức là mỗi hạt nhân (*kernel*) có một lưới (*grid*) 3 chiều (3D) với các phần tử là một khối một chiều (1D) của các *threads*. Nếu ký hiệu

*blockDim.x*, *y*, *z* – số lượng tiểu trình trong một khối, theo hướng tương ứng *x*, *y*, *z*; *GridDim.x*, *y*, *z* – số lượng *blocks* trong một lưới theo hướng tương ứng *x*, *y*, *z*; thì chỉ mục khối được xác định như sau:

$$blockId = blockIdx.x + blockIdx.y * gridDim.x + gridDim.x * gridDim.y * blockIdx.z;$$

Và chỉ mục tiểu trình được tính bằng:

$$threadId = blockId * blockDim.x + threadIdx.x;$$

Khi đó chỉ mục của mỗi *warp* được xác định bằng:

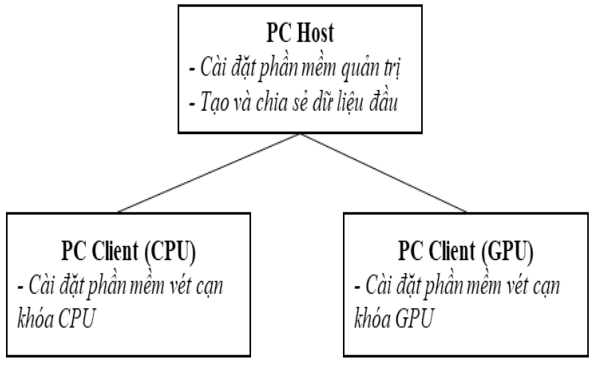
$$warp\_Id = threadId / 32;$$

Việc thử mỗi khóa được tiến hành song song trên các *warp* sau khi tính toán số lượng tiểu trình phù hợp với từng card GPU hay từng khoảng mà máy tính được phân chia.

## III. Thực nghiệm thám mã khối DES

### *A. Xây dựng hệ thống và cài đặt phần mềm.*

Mô hình cài đặt phần mềm phục vụ triển khai thử nghiệm được trình bày như Hình 6. Trong đó sử dụng 1 máy tính đóng vai trò máy quản trị, được cài đặt phần mềm quản trị kết nối vào mạng và chia sẻ dữ liệu thám mã. Đối với các máy trong mạng không có card GPU sẽ được cài đặt và chạy phần mềm vét cạn khóa dành cho CPU. Đối với các máy client có sử dụng card GPU sẽ được cài đặt và chạy phần mềm vét cạn khóa có ứng dụng lập trình song song trên GPU.




Hình 6. Mô hình cài đặt phần mềm trên hệ thống mạng máy tính phân tán.

Dữ liệu phục vụ cho vét cạn trên các máy sẽ được tạo và chia sẻ bởi máy quản trị. Sau khi tạo các dữ liệu và chia sẻ, các máy khác trong mạng tùy vào thời gian rảnh rỗi của mình có thể chạy phần mềm thám mã để thực hiện quá trình thám mã. Kết quả vét cạn trên các máy tham gia này sẽ được tự động gửi về máy quản trị dưới dạng số nguyên 64 bit. Các ký hiệu sử dụng trong phần B:

*Cipher*: bản mã;

*Plaintext*: bản rõ;

*Plaintext length*: Độ dài bản rõ;

*Key*: Khóa mã;

*Key_length*: Độ dài khóa;

*Number_of_interval*: Số lượng khoảng.

### *B. Kết quả thử nghiệm thám mã DES.*

Quá trình thử nghiệm thám mã DES khi biết 1 cặp rõ-mã được tiến hành trên các dữ liệu đầu vào sau:

*Khối bản rõ tương ứng biết trước:* ***cong hoa***

*Cipher: "****1cd59fa9307ddab1****";*

*Plaintext: "****cong hoa****";*

*Plaintext length = ****8****;*

*Key: "****abcd****";*

*Key_length = ****4****;*

*Number_of_interval:* ***50***

Tiến hành lần lượt 2 thử nghiệm với hệ thống gồm 1 máy quản trị và 5 máy client. Trong đó có 2 máy có card GPU. Kết quả thám mã được thể hiện tóm tắt tại Bảng 1 với trường hợp biết trước 1 cặp rõ-mã và không gian các ký tự khóa tăng dần. Bảng 2 thể hiện kết quả thám mã trong trường hợp tăng dần độ dài của khóa mã. Số lượng phép thử được tính bằng tích của số khả năng có thể của khóa với các khả năng có thể của bản rõ.

BẢNG 1. Kết quả tổng hợp thám mã khi biết trước 1 cặp rõ-mã trường hợp tăng không gian ký tự khóa

| STT | Không gian các ký tự khóa | Số lượng phép thử | Thời gian thám mã |
|---|---|---|---|
| 1 | a-z | $\approx 2^{18.8}$ | 0.447 s |
| 2 | A-Z,a-z | $\approx 2^{22.8}$ | 5.8 s |
| 3 | 0-9,A-Z,a-z | $\approx 2^{23.8}$ | 12 s |

BẢNG 2. KẾT QUẢ THÁM MÃ KHI BIẾT TRƯỚC 1 CẶP RÕ-MÃ TRƯỜNG HỢP TĂNG ĐỘ DÀI KHÓA

| STT | Độ dài khóa (ký tự) | Số lượng phép thử | Thời gian thám mã |
|---|---|---|---|
| 1 | 4 | $\approx 2^{18.8}$ | 0.447 s |
| 2 | 5 | $\approx 2^{23.5}$ | 4.4 s |
| 3 | 6 | $\approx 2^{28.2}$ | 1.23 phút |
| 4 | 7 | $\approx 2^{32.9}$ | 17.5 phút |
| 5 | 8 | $\approx 2^{37.6}$ | 3.5 h |

Quá trình thử nghiệm thám mã DES khi chỉ biết bản mã được tiến hành trên các dữ liệu đầu vào sau:

*Plaintext_alphabet:* ***acghno***

*Plaintext_length:* ***8***

*Key_length: 4*

*Key_alphabet:* ***abcdefghijklmnopqrstuvwxyz***

*Number_of_interval:* ***49***

Đối với thử nghiệm này độ dài bản rõ được cố định là 8 ký tự, độ dài khóa sẽ thay đổi lần lượt từ 4 đến 8 ký tự. Không gian các ký tự khóa và bản rõ được giữ nguyên. Kết quả thám mã được thể hiện trong Bảng 3.

BẢNG 3. KẾT QUẢ THÁM MÃ KHI CHỈ BIẾT BẢN MÃ

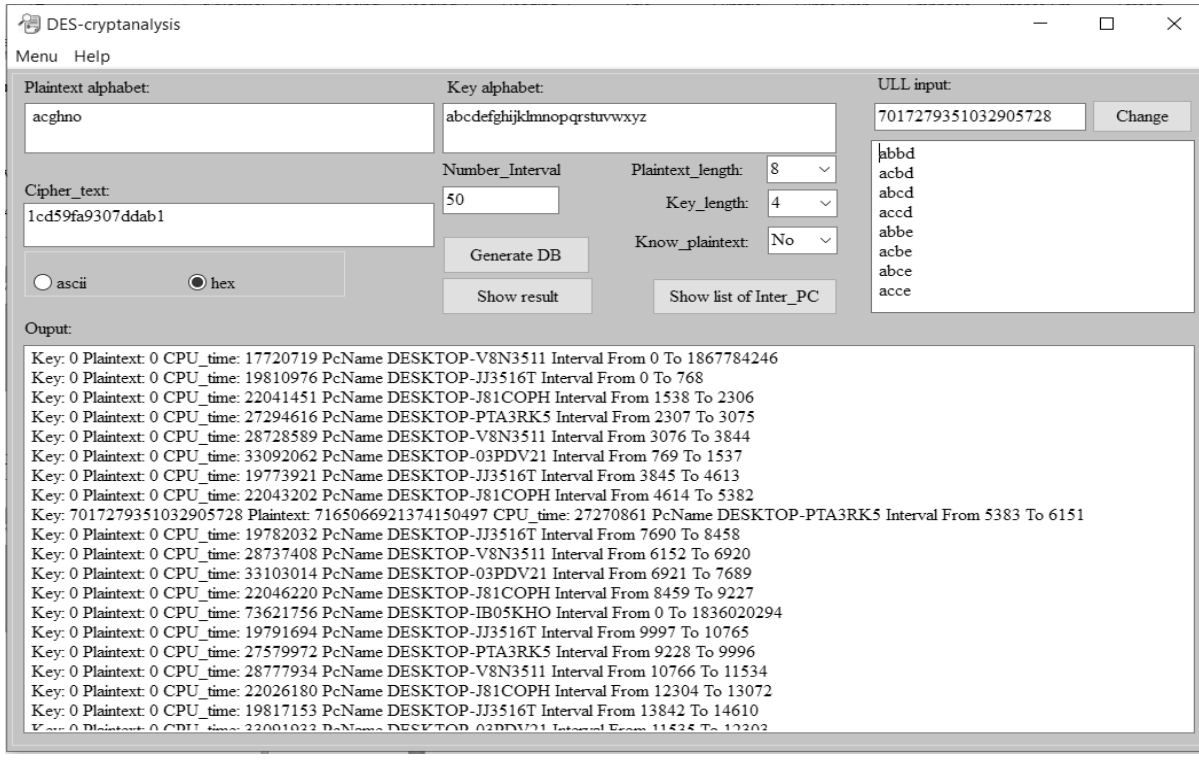
| STT | Độ dài khóa (ký tự) | Tổng số lượng phép thử | Tổng thời gian vét cạn | Số lượng máy tính cần thực hiện thám mã trong 6 tháng (ước tính) |
|---|---|---|---|---|
| 1 | 4 | $\approx 2^{41.25}$ | 15,158 ngày | |
| 2 | 5 | $\approx 2^{45.94}$ | 391,266 ngày (ước tính) | 11 máy |
| 3 | 6 | $\approx 2^{50.66}$ | 10.311,792 ngày (ước tính) | 283 máy |
| 4 | 7 | $\approx 2^{55.37}$ | 269.889,1599 Ngày (ước tính) | 7.390 máy |
| 5 | 8 | $\approx 2^{60.06}$ | 6.966.523,736 ngày (ước tính) | 190.734 máy |

Bảng 4 biểu thị sự so sánh thời gian thám mã đối với máy tính CPU và các máy tính có sử dụng card đồ họa GPU.

BẢNG 4. SO SÁNH THỜI GIAN THÁM MÃ TRÊN MÁY TÍNH CPU VÀ GPU

| Cấu hình máy tính | Số lượng phép thử | Thời gian thám mã |
|---|---|---|
| DESKTOP<br>System: Intel(R) Core(TM) i7, RAM 16.0 GB, System type 64 bit, Window 10 Pro N | $\approx 2^{26.04}$ | 45,4 s |
| DESKTOP<br>System: Intel(R) Core(TM) i7, RAM 16.0 GB, System type 64 bit, Window 10 Pro N,<br>Card GPU: NVIDIA Geforce GT 730 | $\approx 2^{26.04}$ | 4,1 s |

Kết quả thám mã từ các máy client sau khi hoàn thành được gửi về máy quản trị để phân tích và giải mã (Hình 7).



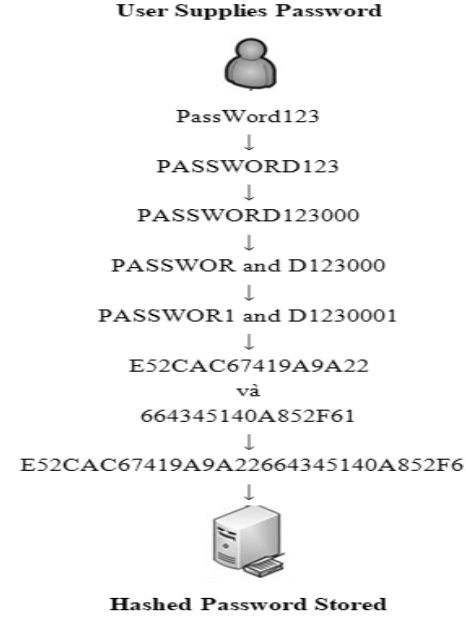

Hình 7. Kết quả phân tích và giải mã khóa trên máy quản trị.

Đối với thám mã vét cạn sử dụng năng lực của hệ thống máy tính phân tán có kết nối mạng. Không gian các ký tự và độ dài của khóa cũng như bản rõ ảnh hưởng trực tiếp tới thời gian thám mã. Không gian và độ dài càng tăng thì thời gian thám mã càng tăng. Để rút ngắn thời gian cần tăng số lượng máy tính client tham gia thám mã. Tốc độ thám mã của máy tính có sử dụng card GPU (Nvidia Geforce GT 730) nhanh gấp khoảng 11 lần so với máy tính có cùng cấu hình khi không sử dụng card GPU.

Quá trình sử dụng bộ công cụ phục vụ cho thám mã ảnh hưởng không đáng kể tới quá trình làm việc của các máy client. Đặc biệt có thể khai thác tối đã năng lực của các máy tính trong thời gian rảnh rỗi.

## IV. THÁM MẬT KHẨU WINDOWS

Windows sử dụng hai phương pháp hash mật khẩu người dùng, cả hai đều có những điểm mạnh và điểm yếu riêng. Đó là LAN Manager (LM) và NT LAN Manager version 2 (NTLMv2). Hàm hash là hàm một chiều mà nếu đưa một lượng dữ liệu bất kì qua hàm này sẽ cho ra một chuỗi có độ dài cố định. LM hash là một trong những thuật toán băm mật khẩu đầu tiên được sử dụng bởi các hệ điều hành Windows, đây là phiên bản duy nhất được hỗ trợ cho tới khi xuất hiện NTLMv2. Hiện nay nó vẫn sử dụng trong các phiên bản Windows 2000, XP, Vista và Windows 7. LM hash được thực hiện qua 6 bước sau (Hình 8):




Hình 8. Quá trình biến đổi mật khẩu thành một LM hash trong Windows.

- Chuyển mật khẩu thành các ký tự in hoa;
- Bổ sung thêm các ký tự 0 vào mật khẩu cho tới khi đủ 14 ký tự;
- Chia mật khẩu mới thành hai phần, mỗi phần có 7 ký tự;
- Tạo khóa DES từ hai nửa trên bằng cách thêm vào một bit chẵn lẻ để tạo các khóa 64 bit;
- Mã hóa một chuỗi ASCII mặc định (**KGS!@#$%**) lần lượt bằng hai khóa được tạo ở trên, cho kết quả ra trong hai chuỗi văn bản mật 8 byte;
- Ghép hai chuỗi văn bản mật 8 byte thu được sau khi mã hóa thành một giá trị 16 byte. Giá trị này chính là một LM hash hoàn chỉnh.

Các mật khẩu tuân theo phương pháp LM hash sử dụng phương pháp mã hóa DES. Điểm mạnh lớn nhất trong LM hash chính là trong quá trình tạo khóa DES. Trong quá trình này, mật khẩu được cấp bởi người dùng sẽ tự động chuyển đổi tất cả thành in hoa, sau đó được chèn thêm thành chuỗi có độ dài 14 ký tự (đây là chiều dài tối đa cho mật khẩu theo phương pháp LM hash), tiếp đó được chia thành hai hash 7 ký tự. Đây có thể coi là một điểm yếu vì chuỗi mật mã bị chia nhỏ và chỉ sử dụng các ký tự ASCII in hoa. Xét về bản chất, thuật toán này làm cho việc sử dụng các ký tự khác cũng như tăng chiều dài mật khẩu trở nên vô nghĩa, đó chính là điều làm cho các mật khẩu LM trở nên dễ bị tấn công bằng phương pháp vét cạn.

Có thể sử dụng nhiều phần mềm để thu được mã băm mật khẩu của windows như Meterpreter. Các hàm băm này sẽ được sử dụng để tìm ra tên người dùng và mật khẩu tương ứng qua. Tệp mật khẩu chứa thông tin tài khoản người dùng được hiển thị như sau:

```
Administrator:500:CEEB0FA9F240C200417EAF40CFAC29C3:D280553F0103F2E643406517296E7582:::
User1:1011:7584248B8D2C9F9EAAD3B435B51404EE:186CB09181E2C2ECAAC768C47C729904:::
User2:1012:AC5BA6A944526699AAD3B435B51404EE:F07A9DFFFC2C5C7F9D9EBC83FD69D68E:::
User3:1013:E7EED3F5C2C85B88AAD3B435B51404EE:6AA15B3D14492D3FA4AA7C5E9CDC0E6A:::
```

Hình 9. Mã băm mật khẩu từ Windows.

Mỗi trường được phân cách với nhau bằng dấu ':' cụ thể như sau:

Trường thứ nhất: Tên đăng nhập (Administrator, User1, etc.).

Trường thứ hai: Mã nhận dạng tương đối (Relative Identification RID): Nhận dạng tương đối (RID): 3-4 chữ số cuối cùng của Mã định danh bảo mật (Security Identifier SID), mỗi người dùng có một mã duy nhất.

Trường thứ ba: LM hash:

```
CEEB0FA9F240C200417EAF40CFAC29C3
```

```
7584248B8D2C9F9EAAD3B435B51404EE
```

```
AC5BA6A944526699AAD3B435B51404EE
```

```
E7EED3F5C2C85B88AAD3B435B51404EE
```

Trường thứ 4: NTLM hash:

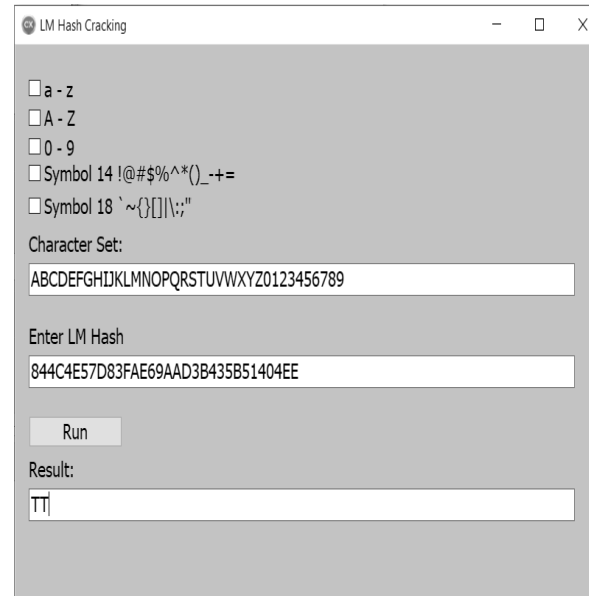
```
D280553F0103F2E643406517296E7582
```

```
186CB09181E2C2ECAAC768C47C729904
```

```
F07A9DFFFC2C5C7F9D9EBC83FD69D68E
```

```
6AA15B3D14492D3FA4AA7C5E9CDC0E6A
```

Để giải mã mật khẩu windows chúng ta sử dụng chức năng LM_Hash_Cracking trên phần mềm quản trị. Giao diện phần bẻ khóa mật khẩu bằng hàm băm LM Hash như Hình 10. Sau khi nhập đầy đủ các trường chúng ta click vào nút "Run" để tiến hành bẻ khóa mật khẩu và chờ kết quả. Kết quả sẽ được hiển thị trên trường Result.




Hình 10. Kết quả thực hiện thám mật khẩu Windows.

## KẾT LUẬN

Ứng dụng mạng máy tính phân tán cho quá trình thám mã giúp tối ưu được việc khai thác sử dụng phần cứng phân tán trong thời gian rảnh rỗi. Kết hợp các máy tính có card GPU sẽ phát huy tối đa hiệu suất và hiệu quả làm việc của từng máy tính trong các cơ quan, đơn vị. Ngoài ra khai thác và sử dụng hệ thống máy tính phân tán bằng bộ công cụ thám mã sẽ mở ra một hướng ứng dụng mới trong việc nghiên cứu thám mã khối nói riêng và tiến tới là các hệ mã khác. Không cần thiết phải đầu tư ngay các trang thiết bị hiện đại, tối ưu nhất. Từ kết quả thám mã đạt được sẽ hỗ trợ cho quá trình giải mã các tập tin bị mã hóa thu thập được cũng như giúp kiểm thử độ an toàn của các phương pháp mã hóa trước khi ứng dụng vào thực tế.

## TÀI LIỆU THAM KHẢO


[1] Ajay D. Kshemkalyani, Mukesh Singhal (2011). Distributed Computing: Principles, Algorithms, and Systems. Cambridge University Press. pp. 736. ISBN 978-0-521-87634-6.

[2] Basu, S. K. (2016). Parallel and distributed computing: architectures and algorithms. PHI Learning Pvt. pp. 242. ISBN 978-81-203-5212-4.

[3] Bátiz-Lazo, Bernardo (2018). Cash and Dash: How ATMs and Computers Changed Banking. Oxford University Press. pp. 284 & 311. ISBN 9780191085574.

[4] Biham, Eli; Dunkelman, Orr; Keller, Nathan (2002-12-01). Enhancing Differential-Linear Cryptanalysis. Advances in Cryptology – ASIACRYPT 2002. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. pp. 254–266. DOI:10.1007/3-540-36178-2_16. ISBN 978 -3540361787.

[5] Biryukov, Alex; Cannière, Christophe De; Quisquater, Michaël (2004-08-15). On Multiple Linear Approximations. Advances in Cryptology – CRYPTO 2004. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. pp. 1–22. DOI:10.1007/978-3-540-28628-8_1. ISBN 9783540226680.

[6] Diffie, Whitfield; Hellman, Martin E. (June 1977). "Exhaustive Cryptanalysis of the NBS Data Encryption Standard" (PDF). Computer. 10 (6): 74–84. DOI:10.1109/C-M.1977.217750. S2CID 2412454. Archived from the original (PDF) on 2014-02-26.

[7] Hoàng Minh Tuấn (2008). "Thám mã khối trên máy tính song song dùng hệ điều hành Linux", Luận văn Tiến sĩ, Học viện Kỹ thuật Quân sự, tr. 215.

[8] Junod, Pascal (2001-08-16). On the Complexity of Matsui's Attack. Selected Areas in Cryptography. Lecture Notes in Computer Science. 2259. Springer, Berlin, Heidelberg. pp. 199–211. DOI:10.1007/3-540-45537-X_16. ISBN 978-3540455370.

[9] Pablo Freyre, Oristela Cuellar, Nelson Díaz, Adrián Alfonso, "Block Ciphers with Matrices Operating Alternately over Columns and Rows", Journal of Science and Technology on Information Security, ISSN 2615-9570, Vol. 12, No. 02, 2020, pp. 18-29.


## SƠ LƯỢC VỀ TÁC GIẢ


**Hoàng Thái Hổ**

Đơn vị công tác: Phòng Thí nghiệm Trọng điểm An toàn thông tin, Bộ Tư lệnh 86.

Email: hoangthaiho@gmail.com

Quá trình đào tạo: Nhận bằng Kỹ sư Bảo đảm toán học và tin học cho kỹ thuật tính toán và hệ thống tự động hóa tại Đại học Kỹ thuật Pháo binh Penza, Liên bang Nga năm 2014; Nhận bằng Tiến sĩ Mô phỏng toán học, phương pháp số và tổ hợp phần mềm tại Đại học Tổng hợp Quốc gia Penza, Liên bang Nga năm 2018.

Hướng nghiên cứu hiện nay: Toán học ứng dụng, mật mã học, công nghệ Blockchain và IoT.

**Nguyễn Thế Hùng**

Đơn vị công tác: Khoa Tên lửa, Học viện Phòng không – Không quân.

Quá trình đào tạo: Tốt nghiệp Kỹ sư, Học viện Kỹ thuật quân sự năm 2001; Thạc sĩ, Học viện Kỹ thuật quân sự, chuyên ngành Điều khiển các thiết bị bay năm 2008.

Hướng nghiên cứu: Bảo mật, điều khiển thiết bị bay.

**Nguyễn Tuấn Minh**

Đơn vị công tác: Trung tâm CNTT, Học viện Ngân hàng.

Quá trình đào tạo: Tốt nghiệp Kỹ sư, Học viện Kỹ thuật Hàng không Moscow (MATI), ngành Công nghệ Thông tin năm 2004; tốt nghiệp Thạc sĩ, Đại học Công nghệ - Đại Học Quốc gia Hà Nội, chuyên ngành Hệ thống Thông tin Quản lý năm 2014.

Hướng nghiên cứu: Khai phá dữ liệu.