# Chương I :Tổng quan

## 1.1 Tổng quan về An ninh mạng

An ninh mạng (cybersecurity), an ninh máy tính (computer security), bảo mật công nghệ thông tin (IT security) là việc bảo vệ hệ thống mạng máy tính khỏi các hành vi [trộm cắp](https://vi.wikipedia.org/wiki/Tr%E1%BB%99m_c%E1%BA%AFp) hoặc làm tổn hại đến phần cứng, phần mềm và các dữ liệu, cũng như các nguyên nhân dẫn đến sự [gián đoạn](https://vi.wikipedia.org/wiki/T%E1%BA%A5n_c%C3%B4ng_t%E1%BB%AB_ch%E1%BB%91i_d%E1%BB%8Bch_v%E1%BB%A5), [chuyển lệch hướng](https://vi.wikipedia.org/wiki/Botnet) của các dịch vụ hiện đang được được cung cấp.

An ninh mạng là thực tiễn của việc bảo vệ các hệ thống điện tử, mạng lưới, máy tính, thiết bị di động, chương trình và dữ liệu khỏi những cuộc tấn công kỹ thuật số độc hại có chủ đích. Tội phạm mạng có thể triển khai một loạt các cuộc tấn công chống lại các nạn nhân hoặc doanh nghiệp đơn lẻ; có thể kể đến như truy cập, làm thay đổi hoặc xóa bỏ dữ liệu nhạy cảm; tống tiền; can thiệp vào các quy trình kinh doanh.

An ninh [mạng máy tính](https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_m%C3%A1y_t%C3%ADnh) bao gồm việc kiểm soát truy cập vật lý đến phần cứng, cũng như bảo vệ chống lại tác hại có thể xảy ra qua truy cập mạng máy tính, cơ sở dữ liệu ([SQL injection](https://vi.wikipedia.org/wiki/SQL_injection)) và việc lợi dụng lỗ hổng phần mềm (code injection). Do sai lầm của những người điều hành, dù cố ý hoặc do bất cẩn, an ninh công nghệ thông tin có thể bị [lừa đảo phi kỹ thuật](https://vi.wikipedia.org/wiki/L%E1%BB%ABa_%C4%91%E1%BA%A3o_phi_k%E1%BB%B9_thu%E1%BA%ADt) để vượt qua các thủ tục an toàn thông qua các phương pháp khác nhau.

An ninh mạng hoạt động thông qua một cơ sở hạ tầng chặt chẽ, được chia thành ba phần chính: bảo mật công nghệ thông tin, an ninh mạng và an ninh máy tính.

Bảo mật công nghệ thông tin (với cách gọi khác là bảo mật thông tin điện tử): Bảo vệ dữ liệu ở nơi chúng được lưu trữ và cả khi các dữ liệu này di chuyển trên các mạng lưới thông tin. Trong khi an ninh mạng chỉ bảo vệ dữ liệu số, bảo mật công nghệ thông tin nắm trong tay trọng trách bảo vệ cả dữ liệu kỹ thuật số lẫn dữ liệu vật lý khỏi những kẻ xâm nhập trái phép.

An ninh mạng: Là một tập hợp con của bảo mật công nghệ thông tin. An ninh mạng thực hiện nhiệm vụ đảm bảo dữ liệu kỹ thuật số trên các mạng lưới, máy tính và thiết bị cá nhân nằm ngoài sự truy cập, tấn công và phá hủy bất hợp pháp.

An ninh máy tính: Là một tập hợp con của an ninh mạng. Loại bảo mật này sử dụng phần cứng và phần mềm để bảo vệ bất kỳ dữ liệu nào được gửi từ [máy tính cá nhân](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh_c%C3%A1_nh%C3%A2n) hoặc các thiết bị khác đến hệ thống mạng lưới thông tin. An ninh máy tính thực hiện chức năng bảo vệ cơ sở hạ tầng công nghệ thông tin và chống lại các dữ liệu bị chặn, bị thay đổi hoặc đánh cắp bởi tội phạm mạng.

Lĩnh vực này dần trở nên quan trọng do sự phụ thuộc ngày càng nhiều vào các hệ thống máy tính và Internet tại các quốc gia, cũng như sự phụ thuộc vào hệ thống mạng không dây như [Bluetooth](https://vi.wikipedia.org/wiki/Bluetooth), [Wi-Fi](https://vi.wikipedia.org/wiki/Wi-Fi), cùng với sự phát triển của các thiết bị "thông minh", bao gồm [điện thoại thông minh](https://vi.wikipedia.org/wiki/%C4%90i%E1%BB%87n_tho%E1%BA%A1i_th%C3%B4ng_minh), TV và các thiết bị khác kết nối vào hệ thống [Internet of Things](https://vi.wikipedia.org/wiki/Internet_of_Things).

Nhân sự làm việc trong mảng an ninh mạng có thể được chia thành 3 dạng sau:

Hacker mũ trắng (White-hat [hacker](https://vi.wikipedia.org/wiki/Hacker_(an_ninh_m%C3%A1y_t%C3%ADnh)))  – cũng còn gọi là "ethical hacker" (hacker có nguyên tắc/đạo đức) hay penetration tester (người xâm nhập thử nghiệm vào hệ thống). Hacker mũ trắng là những chuyên gia công nghệ làm nhiệm vụ [xâm nhập thử nghiệm](https://vi.wikipedia.org/wiki/Ki%E1%BB%83m_tra_th%C3%A2m_nh%E1%BA%ADp) vào hệ thống công nghệ thông tin để tìm ra lỗ hổng, từ đó yêu cầu người chủ hệ thống phải vá lỗi hệ thống để phòng ngừa các xâm nhập khác sau này với ý đồ xấu (thường là của các hacker mũ đen).

Hacker mũ đen (Black-hat hacker): là các chuyên gia công nghệ xâm nhập vào hệ thống với mục đích xấu như đánh cắp thông tin, phá hủy hệ thống, làm lây nhiễm các phần mềm độc hại cũng như các hành vi phá hoại mạng máy tính vi phạm pháp luật khác.

Hacker mũ xám (Grey-hat hacker): là các chuyên gia công nghệ có thể vừa làm công nghệ của cả hacker mũ trắng và mũ đen.

Mục tiêu của an ninh mạng

Mục tiêu của an ninh mạng là bảo vệ thông tin khỏi bị đánh cắp, xâm phạm hoặc bị tấn công. Độ bảo mật an ninh mạng có thể được đo lường bằng ít nhất một trong ba mục tiêu sau:

Bảo vệ tính bảo mật của dữ liệu.

Bảo toàn tính toàn vẹn của dữ liệu.

Thúc đẩy sự sẵn có của dữ liệu cho người dùng được ủy quyền.

Những mục tiêu này tạo thành bộ ba "Bảo mật – Toàn vẹn – Sẵn có" (Confidentiality – Integrity – Availability), đây là cơ sở cốt lõi của tất cả các chương trình bảo mật thông tin. Tam giác CIA là một mô hình bảo mật được thiết kế để hướng dẫn thực thi các chính sách bảo mật thông tin trong khuôn khổ nội bộ một tổ chức hoặc một công ty. Mô hình này cũng được gọi là AIC để tránh sự nhầm lẫn với [Cơ quan Tình báo Trung ương](https://vi.wikipedia.org/wiki/C%C6%A1_quan_T%C3%ACnh_b%C3%A1o_Trung_%C6%B0%C6%A1ng_(Hoa_K%E1%BB%B3)) Hoa Kỳ (CIA - Central Intelligence Agency).

Tiêu chí của CIA được hầu hết các tổ chức và công ty sử dụng khi họ bắt tay vào cài đặt một ứng dụng mới, tạo lập cơ sở dữ liệu hoặc khi muốn đảm bảo quyền truy cập vào một số dữ liệu nói chung. Để dữ liệu được bảo mật hoàn toàn, tất cả các tiêu chí này phải có hiệu lực, đây là những chính sách bảo mật mà mọi thành phần cấu tạo nên nó đều phải cùng nhau hoạt động, và do đó, có thể sẽ xảy ra sai sót khi bỏ quên một trong những thành phần của CIA.

Các yếu tố của tam giác CIA được coi là ba yếu tố quan trọng nhất của bảo mật thông tin.

Tính bảo mật (Confidentiality)

Bảo mật gần tương đương với quyền riêng tư và việc tránh tiết lộ thông tin trái phép. Liên quan đến việc bảo vệ dữ liệu, bảo mật cung cấp quyền truy cập cho những người được phép và ngăn chặn người khác tiếp xúc với bất kỳ thông tin nào về nội dung của chủ sở hữu. Yếu tố này ngăn chặn thông tin cá nhân tiếp cận sai người trong khi đảm bảo rằng người dùng mục tiêu có thể thu thập được thông tin cần thiết. Mã hóa dữ liệu là một ví dụ điển hình để đảm bảo tính bảo mật.

Các công cụ chính phục vụ cho tiêu chí "bảo mật":

Mã hóa (Encryption): Mã hóa là một phương pháp chuyển đổi thông tin khiến dữ liệu trở nên không thể đọc được đối với người dùng trái phép bằng cách sử dụng thuật toán. Sử dụng khóa bí mật (khóa mã hóa) để dữ liệu được chuyển đổi, chỉ có thể được đọc bằng cách sử dụng một khóa bí mật khác (khóa giải mã). Công cụ này nhằm bảo vệ những dữ liệu nhạy cảm như số [thẻ tín dụng](https://vi.wikipedia.org/wiki/Th%E1%BA%BB_t%C3%ADn_d%E1%BB%A5ng), bằng cách mã hóa và chuyển đổi dữ liệu thành một văn bản mật mã không thể đọc được, dữ liệu này chỉ có thể được đọc một khi đã giải mã nó. Khóa bất đối xứng (asymmetric-key) và khóa đối xứng (symmetric-key) là hai loại mã hóa chính phổ biến nhất.

Kiểm soát quyền truy cập (Access Control): Đây là công cụ xác định các quy tắc và chính sách để giới hạn quyền truy cập vào hệ thống hoặc các tài nguyên, dữ liệu ảo/vật lý. Kiểm soát quyền truy cập bao gồm quá trình người dùng được cấp quyền truy cập và một số đặc quyền nhất định đối với hệ thống, tài nguyên hoặc thông tin. Trong các hệ thống kiểm soát quyền truy cập, người dùng cần xuất trình thông tin đăng nhập trước khi có thể được cấp phép tiếp cận thông tin, có thể kể đến như danh tính, số seri của máy chủ. Trong các hệ thống vận hành vật lý, các thông tin đăng nhập này có thể tồn tại dưới nhiều dạng, nhưng với các thông tin không thể được chuyển giao sẽ cung cấp tính bảo mật cao nhất.

Xác thực (Authentication): Xác thực là một quá trình đảm bảo và xác nhận danh tính hoặc vai trò của người dùng. Công cụ này có thể được thực hiện theo một số cách khác nhau, nhưng đa số thường dựa trên sự kết hợp với: một thứ gì đó mà cá nhân sở hữu (như thẻ thông minh hoặc khóa radio để lưu trữ các khóa bí mật), một thứ gì đó mà cá nhân biết (như mật khẩu) hoặc một thứ gì đó dùng để nhận dạng cá nhân (như dấu vân tay). Xác thực đóng vai trò cấp thiết đối với mọi tổ chức, vì công cụ này cho phép họ giữ an toàn cho mạng lưới thông tin của mình bằng cách chỉ cho phép người dùng được xác thực truy cập vào các tài nguyên dưới sự bảo vệ, giám sát của nó. Những tài nguyên này có thể bao gồm các hệ thống máy tính, mạng, cơ sở dữ liệu, website và các ứng dụng hoặc dịch vụ dựa trên mạng lưới khác.

Cấp quyền (Authorization): Đây là một cơ chế bảo mật được sử dụng để xác định quyền hạn (privilege) một người nào đó đối với các tài nguyên như các chương trình máy tính, tệp tin, dịch vụ, dữ liệu và tính năng ứng dụng. Ủy quyền thường được đi sau xác thực nhằm xác định một user sau khi đã đăng nhập thành công thì được phép làm những gì đối với từng loại dữ liệu. [Quản trị viên hệ thống](https://vi.wikipedia.org/wiki/Qu%E1%BA%A3n_tr%E1%BB%8B_vi%C3%AAn_h%E1%BB%87_th%E1%BB%91ng) thường là người chỉ định cấp phép hoặc từ chối quyền truy cập đối với cá nhân khi muốn tiếp cận thông tin dữ liệu và đăng nhập vào hệ thống.

Bảo mật vật lý (Physical Security): Đây là các biện pháp được thiết kế để ngăn chặn sự truy cập trái phép vào các tài sản công nghệ thông tin như cơ sở vật chất, thiết bị, nhân sự, tài nguyên và các loại tài sản khác nhằm tránh bị hư hại. Công cụ này bảo vệ các tài sản nêu trên khỏi các mối đe dọa vật lý như: trộm cắp, phá hoại, hỏa hoạn và thiên tai.

Tính toàn vẹn (Integrity)

Tính toàn vẹn đề cập đến các phương pháp nhằm đảm bảo nguồn dữ liệu là thật, chính xác và được bảo vệ khỏi sự sửa đổi trái phép của người dùng.

Các công cụ chính phục vụ cho tiêu chí "toàn vẹn":

Sao lưu (Backups): Sao lưu là lưu trữ dữ liệu định kỳ. Đây là một quá trình tạo lập các bản sao của dữ liệu hoặc tệp dữ liệu để sử dụng trong trường hợp khi dữ liệu gốc hoặc tệp dữ liệu bị mất hoặc bị hủy. Sao lưu cũng được sử dụng để tạo các bản sao phục vụ cho các mục đích lưu lại lịch sử dữ liệu, chẳng hạn như các nghiên cứu dài hạn, thống kê hoặc cho các ghi chép, hoặc đơn giản chỉ để đáp ứng các yêu cầu của chính sách lưu trữ dữ liệu.

Tổng kiểm tra (Checksums): Tổng kiểm tra là một giá trị số được sử dụng để xác minh tính toàn vẹn của tệp hoặc dữ liệu được truyền đi. Nói cách khác, đó là sự tính toán của một hàm phản ánh nội dung của tệp thành một giá trị số. Chúng thường được sử dụng để so sánh hai bộ dữ liệu, nhằm đảm bảo rằng chúng giống hệt nhau. Hàm tổng kiểm tra phụ thuộc vào toàn bộ nội dung của tệp, nó được thiết kế theo cách mà ngay cả một thay đổi nhỏ đối với tệp đầu vào (chẳng hạn như lệch một bit) có thể dẫn đến giá trị đầu ra khác nhau.

Mã chỉnh dữ liệu (Data Correcting Codes): Đây là một phương pháp để lưu trữ dữ liệu theo cách mà những thay đổi nhỏ nhất cũng có thể dễ dàng được phát hiện và tự động điều chỉnh.

Tính sẵn có (Availability)

Mọi hệ thống thông tin đều phục vụ cho mục đích riêng của nó và thông tin phải luôn luôn sẵn sàng khi cần thiết. Hệ thống có tính sẵn sàng cao hướng đến sự sẵn có, khả dụng ở mọi thời điểm, tránh được rủi ro, đảm bảo thông tin có thể được truy cập và sửa đổi kịp thời bởi những người được ủy quyền.

Các công cụ chính phục vụ cho tiêu chí "sẵn có":

Bảo vệ vật lý (Physical Protection): Có nghĩa là giữ thông tin có sẵn ngay cả trong trường hợp phải đối mặt với thách thức về vật chất. Đảm bảo các thông tin nhạy cảm và công nghệ thông tin quan trọng được lưu trữ trong các khu vực an toàn.

Tính toán dự phòng (Computational Redundancies): Được áp dụng nhằm bảo vệ máy tính và các thiết bị được lưu trữ, đóng vai trò dự phòng trong trường hợp xảy ra hỏng hóc.

Lỗ hổng bảo mật và các loại tấn công phổ biến

Lỗ hổng bảo mật là một điểm yếu của hệ thống trong quá trình thiết kế, thi công và quản trị. Phần lớn các lỗ hổng bảo mật được đã phát hiện ngày nay đều được ghi lại trong cơ sở dữ liệu [Common Vulnerabilities and Exposures](https://vi.wikipedia.org/w/index.php?title=Common_Vulnerabilities_and_Exposures&action=edit&redlink=1) (CVE). Một lỗ hổng bị khai thác là một lỗ hổng mà đã bị lợi dụng để thực hiện hoạt động tấn công ít nhất một lần hoặc đã bị khai thác ([exploit](https://vi.wikipedia.org/wiki/Exploit_(an_ninh_m%E1%BA%A1ng))).[[10]](https://vi.wikipedia.org/wiki/An_ninh_m%E1%BA%A1ng#cite_note-10)

Để đảm bảo một hệ thống máy tính, điều quan trọng là phải hiểu các cuộc tấn công có thể được thực hiện chống lại nó, và các mối đe dọa thường được xếp vào một trong các mục dưới đây:

Tấn công bằng [phần mềm độc hại](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_%C3%A1c_%C3%BD) (Malware Attack)

Malware là phần mềm độc hại, được kết hợp giữa hai từ "malicious" và "software". Đây là một trong những hình thức đe dọa mạng phổ biến nhất. Tội phạm mạng và các hacker tạo ra malware với mục đích làm phá vỡ hoặc hư hỏng máy tính của người dùng hợp pháp. Thông thường, hacker sẽ tấn công người dùng thông qua các lỗ hổng bảo mật, dụ dỗ người dùng click vào một đường link đính kèm trong thư rác hoặc tải các tệp tin được ngụy trang hợp-pháp để phần mềm độc hại tự động cài đặt vào máy tính. Malware thường được sử dụng nhằm phục vụ cho mục đích kiếm tiền hoặc tham gia vào các cuộc tấn công mạng có động cơ chính trị.

Có vô số loại phần mềm độc hại khác nhau, điển hình như:

Virus: Là những đoạn mã chương trình tự sao chép, tức tự nhân bản, đính kèm vào các tệp tin sạch, được thiết kế để xâm nhập, lây lan khắp hệ thống máy tính nhằm thực thi một số tác vụ nào đó với nhiều mức độ phá hủy khác nhau.

Trojan Horse: Khác với virus, phần mềm này không có chức năng tự sao chép nhưng lại sở hữu sức công phá tương đương. Trojan Horse sẽ được ngụy trang thành các phần mềm hợp pháp, vô hại, tiếp sau đó, bọn tội phạm mạng lừa người dùng cài đặt Trojan Horse vào máy tính của họ, nơi chúng có thể gây thiệt hại đến máy chủ hoặc thu thập các dữ liệu cá nhân.

[Phần mềm gián điệp](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_gi%C3%A1n_%C4%91i%E1%BB%87p) (Spyware): Đây là loại virus có khả năng thâm nhập trực tiếp vào hệ điều hành mà không để lại "di chứng", bí mật lưu lại những gì người dùng làm, dựa vào đó, tội phạm mạng có thể sử dụng các thông tin này để đem đến bất lợi cho chủ sở hữu chúng.

Phần mềm tống tiền (Ransomware): Ngăn cản người dùng truy cập vào một file hoặc folder quan trọng, chỉ khôi phục với mức tiền chuộc tương ứng.

Phần mềm quảng cáo (Adware): Có thể được sử dụng để phát tán, cài đặt các phần mềm độc hại khác.

Botnets: Mạng lưới các máy tính bị nhiễm phần mềm độc hại được bọn tội phạm mạng sử dụng để thực hiện các tác vụ trực tuyến mà không có sự cho phép của người dùng.

Tấn công giả mạo (Phishing Attack)

Phishing là hình thức giả mạo thành một đơn vị/cá nhân uy tín để chiếm lấy lòng tin của người dùng, với mục tiêu nhắm đến việc đánh cắp dữ liệu cá nhân nhạy cảm như thông tin thẻ tín dụng, mật khẩu, tài khoản đăng nhập hoặc cài đặt các phần mềm độc hại vào máy tính nạn nhân. Phishing thường được thực hiện bằng cách sử dụng [thư điện tử](https://vi.wikipedia.org/wiki/Th%C6%B0_%C4%91i%E1%BB%87n_t%E1%BB%AD) (email) hoặc tin nhắn.

Tấn công trung gian ([Man-in-the-middle Attack](https://vi.wikipedia.org/wiki/T%E1%BA%A5n_c%C3%B4ng_xen_gi%E1%BB%AFa))

Tấn công trung gian (MitM), hay còn gọi là tấn công nghe lén, xảy ra khi kẻ tấn công mạng xâm nhập vào một giao dịch đang diễn ra giữa 2 đối tượng, một khi đã xen vào thành công, chúng có thể chắt lọc và đánh cắp dữ liệu. Một số biến thể của tấn công trung gian có thể kể đến như đánh cắp mật khẩu, chuyển tiếp các thông tin không xác thực. Thông thường, khi sử dụng Wi-Fi công cộng thiếu bảo mật, kẻ tấn công có thể tự "chen" vào giữa thiết bị của người truy cập và mạng Wi-Fi đó, tất cả [dữ liệu cá nhân](https://vi.wikipedia.org/wiki/D%E1%BB%AF_li%E1%BB%87u_c%C3%A1_nh%C3%A2n) mà nạn nhân gửi đi sẽ rơi vào tay bọn tội phạm mà không hề hay biết. Trong một số trường hợp, kẻ tấn công còn cài đặt các ứng dụng khác nhằm thu thập thông tin về nạn nhân thông qua phần mềm độc hại (malware).

Tấn công từ chối dịch vụ (Denial of Service)

Các cuộc tấn công từ chối dịch vụ (DoS) được thiết kế để làm cho tài nguyên mạng hoặc máy không sẵn sàng để phục vụ cho người dùng dự định của nó. [5] Kẻ tấn công có thể từ chối dịch vụ cho từng nạn nhân, chẳng hạn như cố tình nhập sai mật khẩu đủ lần liên tục để khiến tài khoản nạn nhân bị khóa hoặc chúng có thể làm quá tải khả năng của máy hoặc mạng và chặn tất cả người dùng cùng một lúc. Mặc dù một cuộc tấn công mạng từ một [địa chỉ IP](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%8Ba_ch%E1%BB%89_IP) duy nhất có thể bị chặn bằng cách thêm quy tắc tường lửa mới, nhiều hình thức tấn công từ chối dịch vụ phân tán (DDoS) là có thể, trong đó cuộc tấn công đến từ một số lượng lớn máy tính - và việc bảo vệ khó khăn hơn nhiều. Các cuộc tấn công như vậy có thể bắt nguồn từ các máy tính zombie của botnet, nhưng một loạt các kỹ thuật khác có thể bao gồm các cuộc tấn công phản xạ và khuếch đại, trong đó các hệ thống vô tội bị lừa gửi dữ liệu đến máy nạn nhân.

Tấn công cơ sở dữ liệu (SQL Injection Attack)

Hacker chèn một đoạn code độc hại vào server sử dụng ngôn ngữ truy vấn có cấu trúc (SQL), mục đích là để khiến máy chủ trả về những thông tin quan trọng mà lẽ ra không được tiết lộ. Các cuộc tấn công SQL Injection xuất phát từ lỗ hổng của website, hacker có thể tấn công đơn giản bằng cách chèn một đoạn mã độc vào thanh công cụ "Tìm kiếm" là đã có thể dễ dàng tấn công những website với mức bảo mật yếu.

Tấn công "cửa hậu" (Backdoor Attack)

Trong một hệ thống máy tính, Backdoor ("cửa hậu") là một phương pháp bí mật vượt qua thủ tục chứng thực người dùng thông thường hoặc để giữ đường truy nhập từ xa tới một máy tính, trong khi cố gắng không bị phát hiện bởi việc giám sát thông thường. Chúng tồn tại vì một số lý do, bao gồm từ thiết kế ban đầu hoặc từ cấu hình kém. Chúng có thể đã được thêm vào bởi một nhóm có thẩm quyền để cho phép một số truy cập hợp pháp, hoặc bởi những kẻ tấn công vì lý do độc hại; nhưng bất kể động cơ đưa tới sự tồn tại của chúng, chúng tạo ra một lỗ hổng.

Khai thác lỗ hổng (Zero-day Exploits)

Lỗ hổng Zero-day (hay còn gọi là 0-day) là thuật ngữ để chỉ những lỗ hổng phần mềm hoặc phần cứng chưa được biết đến và chưa được khắc phục. Các hacker có thể tận dụng lỗ hổng này để tấn công xâm nhập vào hệ thống máy tính của doanh nghiệp, tổ chức nhằm đánh cắp hoặc thay đổi dữ liệu

## 1.2 Các kiến thức cơ sở

### 1.2.1 Số học Trên Modulo

Trong điện toán, phép toán modulo là phép toán tìm số dư của phép chia 2 số (đôi khi được gọi là modulus).

Cho hai số dư, (số bị chia) a và (số chia) n, a modulo n (viết tắt là a mod n) là số dư của phép chia có dư Euclid của a cho n. Ví dụ, biểu thức "5 mod 2" bằng 1 vì 5 chia cho 2 có thương số là 2 là số dư là 1, trong khi "9 mod 3" bằng 0 do 9 chia 3 có thương số là 3 và số dư 0; không còn gì trong phép trừ của 9 cho 3 nhân 3. (Lưu ý rằng thực hiện phép chia bằng máy tính cầm tay sẽ không hiển thị kết quả giống như phép toán này; thương số sẽ được biểu diễn dưới dạng phần thập phân.)

Mặc dù thường được thực hiện khi a và n đều là số nguyên, nhiều hệ tính toán cho phép sử dụng các kiểu khác của toán học bằng số. Giới hạn của một modulo nguyên của n là từ 0 đến n − 1. (a mod 1 luôn bằng 0; a mod 0 là không xác định, có thể trả về lỗi chia cho số 0 trong nhiều ngôn ngữ lập trình.) Xem số học mô-đun để tìm các quy ước cũ hơn và liên quan được áp dụng trong lý thuyết số.

Số học trên modulo là một nhánh của lý thuyết số, tập trung vào các phép toán và tính chất của các số trong một hệ thống modulo nhất định. Dưới đây là những kiến thức cơ bản về số học modulo:

#### 1.2.1.1 Khái niệm cơ bản

Đồng dư (Congruence): Hai số nguyên a và b được gọi là đồng dư modulo n nếu hiệu của chúng chia hết cho n.

Ký hiệu: 𝑎 ≡ 𝑏(𝑚𝑜𝑑𝑛)

Ví dụ: 17≡5(mod 12) vì 17−5=12 chia hết cho 12.

Phần dư (Remainder): Khi chia một số a cho n, phần dư là số nguyên r sao cho:

𝑎 = 𝑛𝑞 + 𝑟 𝑣ớ𝑖 0 ≤ 𝑟 < 𝑛

#### 1.2.1.2 Tính chất cơ bản

Tính chất tương thích của phép cộng và phép nhân:

Nếu 𝑎 ≡ 𝑏 (𝑚𝑜𝑑 𝑛) và 𝑐 ≡ 𝑑 thì:

𝑎 + 𝑐 ≡ 𝑏 + 𝑑(𝑚𝑜𝑑𝑛)

𝑎 − 𝑐 ≡ 𝑏 − 𝑑(𝑚𝑜𝑑𝑛)

𝑎𝑐 ≡ 𝑏𝑑(𝑚𝑜𝑑𝑛)

#### 1.2.1.3 Phép toán modulo

Cộng modulo: (𝑎 + 𝑏)𝑚𝑜𝑑  𝑛

Trừ modulo: (𝑎 − 𝑏)𝑚𝑜𝑑  𝑛

Nhân modulo: (𝑎 ∗ 𝑏)𝑚𝑜𝑑  𝑛

Lũy thừa modulo: ()𝑚𝑜𝑑  𝑛

**1.2.1.4 Định lý cơ bản**

Định lý Euler: Nếu a và n là hai số nguyên dương và nguyên tố cùng nhau

(không có ước chung ngoài 1), thì ≡ 1(𝑚𝑜𝑑 𝑛) với 𝜙(𝑛) là hàm Euler.

Định lý Fermat nhỏ: Nếu p là một số nguyên tố và a là một số nguyên không chia hết cho p, thì ≡ 1(𝑚𝑜𝑑 𝑝)

**1.2.1.5 Nghịch đảo modulo**

Nghịch đảo của a modulo n là số nguyên b sao cho ab≡1(mod n). Điều kiện tồn tại: a và n phải số nguyên tố cùng nhau.

**1.2.2 Số nguyên tố**

**1.2.2.1 Định nghĩa**

Số nguyên tố: Một số nguyên dương p lớn hơn 1 chỉ có hai ước số dương là 1 và chính nó. Nói cách khác, nếu p là số nguyên tố, thì không có số nguyên dương nào khác ngoài 1 và p chia hết cho p.

Ví dụ: 2, 3, 5, 7, 11, 13, 17, 19, 23, ...

Số hợp (Composite number): Một số nguyên dương lớn hơn 1 không phải là số nguyên tố. Nói cách khác, một số hợp có ít nhất một ước số dương khác ngoài 1 và chính nó.

Ví dụ: 4, 6, 8, 9, 10, 12, 14, 15, 16, ...

**1.2.2.2 Tính chất của số nguyên tố**

Số nguyên tố nhỏ nhất là 2, và cũng là số nguyên tố chẵn duy nhất.

Mọi số nguyên tố lớn hơn 2 đều là số lẻ.

Nếu p là số nguyên tố và p chia hết cho tích ab, thì p phải chia hết cho ít nhất một trong hai số a hoặc b. Điều này được biểu diễn dưới dạng: Nếu p | ab, thì p | a hoặc p | b.

Mọi số nguyên dương lớn hơn 1 đều có thể được phân tích duy nhất thành tích các số nguyên tố (định lý cơ bản của số học).

Ví dụ: 60 = × 3 × 5

**1.2.2. Tính chất liên quan đến các bài toán**

Số nguyên tố tương đối (Hai số nguyên tố cùng nhau): Hai số nguyên a và b là nguyên tố cùng nhau nếu chúng không có ước số chung nào lớn hơn 1, tức là GCD(a, b) = 1.

Định lý Fermat nhỏ: Nếu p là số nguyên tố và a là số nguyên không chia hết cho p, thì a(p−1) ≡ 1(mod p)

Định lý Euler: Tổng quát hóa định lý Fermat nhỏ. Nếu a và n là hai số nguyên dương và coprime, thì a(ϕ(n)) ≡ 1(mod n) với ϕ(n) là hàm Euler, đếm số các số nguyên dương nhỏ hơn n và nguyên tố cùng nhau với n.

**1.2.2.4 Một số số nguyên tố đặc biệt**

Số nguyên tố Mersenne: Là số nguyên tố có dạng − 1 với p cũng là số nguyên tố.

Ví dụ: − 1 = 7

Số nguyên tố Fermat: Là số nguyên tố có dạng + 1.

Ví dụ: 2{} + 1 = 3,2{} + 1 = 5.

**1.2.2.5 Một số thuật toán liên quan đến số nguyên tố**

Thuật toán kiểm tra tính nguyên tố cơ bản:

Kiểm tra chia hết từ 2 đến .

Thuật toán sàng Eratosthenes: Dùng để tìm tất cả các số nguyên tố nhỏ hơn một số nguyên dương N.

Bắt đầu với danh sách các số từ 2 đến N.

Loại bỏ các bội số của từng số nguyên tố bắt đầu từ 2.

Thuật toán kiểm tra tính nguyên tố Miller-Rabin: Một thuật toán ngẫu nhiên, kiểm tra tính nguyên tố của một số lớn.

**1.2.3 Ước số chung lớn nhất**

Ước số chung lớn nhất (Greatest Common Divisor - GCD) là một khái niệm quan trọng trong lý thuyết số và toán học cơ bản. Dưới đây là những khái niệm và tính chất chính liên quan đến GCD.

**1.2.3.1 Định nghĩa**

Ước số chung lớn nhất (GCD): GCD của hai số nguyên a và b, ký hiệu là GCD(a, b) hoặc GCD(a,b) là số nguyên dương lớn nhất mà cả a và b cùng chia hết.

Ví dụ: gcd(48,18)=6 vì 6 là số lớn nhất mà cả 48 và 18 cùng chia hết.

**1.2.3.2 Tính chất của GCD**

Tính chất chia hết: Nếu d là ước số của cả a và b, thì d cũng là ước số của mọi tổ hợp tuyến tính của a và b, tức là 𝑎𝑥 + 𝑏𝑦 sẽ chia hết cho 𝑑 với 𝑥, 𝑦 là các số nguyên.

Tính chất giao hoán:

𝑔𝑐𝑑(𝑎, 𝑏) = 𝑔𝑐𝑑(𝑏, 𝑎)

Tính chất kết hợp:

𝑔𝑐𝑑(𝑎, 𝑔𝑐𝑑(𝑏, 𝑐)) = 𝑔𝑐𝑑(𝑔𝑐𝑑(𝑎, 𝑏), 𝑐)

Tính chất phân phối với phép nhân:

𝑔𝑐𝑑(𝑎, 𝑏𝑐) = 𝑔𝑐𝑑(𝑔𝑐𝑑(𝑎, 𝑏), 𝑔𝑐𝑑(𝑎, 𝑐))

Liên hệ với ước số của hiệu:

𝑔𝑐𝑑(𝑎, 𝑏) = 𝑔𝑐𝑑(𝑎, 𝑏 − 𝑎)𝑏𝑎 ≤ 𝑏

1.2.3.3 Phương pháp tính GCD

Phân tích thừa số nguyên tố:

Phân tích cả hai số thành tích các thừa số nguyên tố.

GCD là tích của các thừa số nguyên tố chung với số mũ nhỏ nhất.

Thuật toán Euclid:

Dựa trên tính chất: 𝑔𝑐𝑑(𝑎, 𝑏) = 𝑔𝑐𝑑(𝑏, 𝑎 𝑚𝑜𝑑 𝑏)

Lặp lại cho đến khi một trong hai số bằng 0.

**1.2.3.4 Ứng dụng của GCD**

Rút gọn phân số: GCD được dùng để rút gọn phân số 𝑎 bằng cách chia cả tử số và mẫu số cho GCD(a, b).

Kiểm tra tính số nguyên tố cùng nhau: Hai số a và b là nguyên tố cùng nhau nếu và chỉ nếu gcd(a,b)=1.

Lý thuyết số: GCD là nền tảng của nhiều khái niệm và định lý trong lý thuyết số, như định lý Euler, định lý Fermat nhỏ.

Giải phương trình Diophantine: Sử dụng GCD để tìm nghiệm nguyên của các phương trình dạng ax+by=c

**1.2.3.5 Tổng kết**

Ước số chung lớn nhất là một công cụ mạnh mẽ trong toán học, giúp giải quyết nhiều bài toán từ cơ bản đến phức tạp. Việc hiểu rõ các tính chất và phương pháp tính GCD không chỉ giúp giải các bài toán số học mà còn mở ra nhiều ứng dụng trong các lĩnh vực khác.

**1.2.4 Phần tử nghịch đảo trên vành Zn**

**1.2.4.1 Thuật toán Euclid mở rộng**

Thuật toán Euclid là một thuật toán để xác định ước chung lớn nhất (GCD – Greatest Common Divisor) của 2 phần tử thuộc vùng Euclid (ví dụ: các số nguyên). Thuật toán Euclid là một trong những thuật toán cổ nhất được biết đến, từ khi xuất hiện trong cuốn Euclid’s Elements khoảng năm 300 trước công nguyên. Euclid khởi đầu đã trình bày rõ ràng vấn đề về phương diện hình học, như vấn đề tìm ra một thước đo chung có độ dài hai đường thẳng, và thuật toán của ông đã xử lý bằng cách lặp lại phép trừ đoạn dài hơn cho đoạn ngắn hơn. Tuy nhiên, thuật toán đã hầu như không được phát hiện ra bởi Euclid và nó đã có thể được biết đến sớm hơn 200 năm. Nó cũng đã được biết đến bởi Eudoxus of Cnidus (khoảng 375 trước công nguyên) và Aristotle (khoảng 330 trước công nguyên).

Giải thuật Euclid mở rộng kết hợp quá trình tìm gcd(a, b) trong thuật toán Euclid với việc tìm một cặp số x, y thoả mãn phương trình Đi-ô-phăng. Giả sử cho hai số tự nhiên a, b, ngoài ra a >b>0. Đặt r0 = a, = b, chia cho được số dư và thương số nguyên q1. Nếu r2 = 0 thì dừng lại, nếu khác không, chia cho được số dư .... Vì dãy các ri là giảm thực sự nến sau hữu

hạn bước ta được số dư = = × + , với

0 <<;

=×+, với

0 < < ;

....;

− 1 = × + + 10 < + 1<r

= + 1 × + 1

trong đó số dư cuối cùng khác 0 là = d. Bài toán đặt ra là tìm x, y sao

cho

a × x + b × y = (= d)

Khi i = n − 1 ta có được và .

**1.2.4.2 Áp dụng giải thuật Euclid mở rộng tìm số nghịch đảo trong vành Zn**

Số nghịch đảo trong vành Zn :

Trong lý thuyết số, vành Zn được định nghĩa là vành thương của với quan hệ đồng theo môđun n (là quan hệ tương đương) mà các phần tử của nó là các lớp đồng dư theo mođun n (n là số nguyên dương lớn hơn 1). Ta cũng có thể xét Zn chỉ với các đại diện của nó. Khi đó

Zn = {0, 1, ........, n – 1}

thông thường được rút gọn theo mođun n:

a + b = (a + b) mod n

a ×b = (a × b) mod n

Phần tử a của Zn được gọi là khả nghịch trong Zn hay khả nghịch theo mođun m nếu tồn tại phần tử a' trong Zn sao cho a × a' = 1 trong Zn hay a × a’ ≡ 1(mod n). Khi đó a' được gọi là nghịch đảo modulo n của a. Trong lý thuyết số đã chứng minh rằng, số a là khả nghịch theo mođun n khi và chỉ khi GCD của a và n bằng 1.

Khi đó tồn tại các số nguyên x, y sao cho n × x + a × y = 1. Đẳng thức này lại chỉ ra x là nghịch đảo của a theo mođun n. Do đó có thể tìm được phần tử nghịch đảo của a theo mođun n nhờ thuật toán Euclid mở rộng khi chia n cho a.

**1.2.5 Thuật toán bình phương và nhân**

Ý nghĩa thuật toán: Thuật toán bình phương và nhân là thuật toán tính

nhanh luỹ thừa tự nhiên của một số (thực hoặc nguyên), trong trường hợp

cơ số là một số nguyên có thể được rút gọn theo một modun nào đó.

Phép nâng lên lũy thừa tự nhiên bậc n của số x (x được gọi là cơ số) được

định nghĩa như sau: ( n chữ số x )

Với n là số lớn, khi đó phép nhân là rất lớn.

**1.2.6 Định lý Euler**

Định lý Euler, một trong những lý thuyết quan trọng trong lý thuyết số, được phát biểu như sau: nếu 𝑛 là một số nguyên dương và 𝑎 là số nguyên tố cùng nhau với 𝑛, thì 𝑎𝜑(𝑛) ≡ 1 (𝑚𝑜𝑑 𝑛), trong đó 𝜑(𝑛) là phi hàm

Euler, đếm số các số nguyên tố cùng nhau với 𝑛 từ 1 đến 𝑛.

Tính chất của hàm Euler.

Nếu 𝑝 là số nguyên tố, thì 𝜑(𝑝) = 𝑝 − 1.

Nếu 𝑝 và 𝑞 là hai số nguyên tố khác nhau, thì 𝜑(𝑝×𝑞) = 𝜑(𝑝) × 𝜑(𝑞).

Nếu là lũy thừa của số nguyên tố 𝑝 với k ≥ 1, thì 𝜑() = − − 1

Trong thuật toán RSA, việc chọn khóa công khai và khóa riêng tư phụ thuộc vào tính chất của phi hàm Euler. Khóa công khai là một cặp số (n, e) trong đó n là tích của hai số nguyên tố và e là số nguyên tố cùng nhau với φ(n). Khóa riêng tư d được tính bởi 𝑑 ≡ 𝑒−1(𝑚𝑜𝑑 𝜑(𝑛)).

**1.2.7 Hàm phi Euler**

Phi hàm Euler, ký hiệu là 𝜙(𝑛), là một hàm toán học đặc biệt trong lý thuyết số, cho biết số các số nguyên dương nhỏ hơn hoặc bằng 𝑛 mà nguyên tố cùng nhau với 𝑛.

Công thức tổng quát: Để tính 𝜙(𝑛) cho một số nguyên 𝑛 =, ta sử dụng công thức:

𝜙(𝑛) =

Tính tổng hàm phi: Một tính chất thú vị của hàm phi Euler là tổng các giá trị hàm phi của tất cả các ước số của 𝑛 bằng 𝑛.

Ứng dụng trong mã hóa: Định lý Euler là nền tảng của hệ thống mã RSA, một phương pháp mã hóa quan trọng được sử dụng rộng rãi trong giao tiếp trên Internet. Sự bảo mật của hệ thống RSA dựa trên độ khó của việc phân tích thừa số của một số lớn thành các số nguyên tố, một vấn đề mà Định lý Euler giúp giải quyết một phần.

**1.2.8 Định lý Fermat nhỏ**

Định lý Euler là một tổng quát hóa của định lý nhỏ Fermat. Khi 𝑛 là số nguyên tố, 𝜑(𝑛)=𝑛−1, và định lý Euler trở thành định lý nhỏ Fermat.

Định lý nhỏ Fermat: Định lý này phát biểu rằng nếu p là một số nguyên tố, thì cho mọi số nguyên a không chia hết cho p, ta có ap - 1≡ 1(mod 𝑝). Đây là trường hợp đặc biệt của Định lý Euler khi n là số nguyên tố.

**1.2.9 Thuật toán kiểm tra số nguyên tố**

Để kiểm tra số nguyên tố, có nhiều thuật toán khác nhau như:

**1.2.9.1 Thuật toán Fermat**

Mô tả: Thuật toán này dựa trên định lý Fermat nhỏ, nếu n là số nguyên tố

thì với bất kỳ số nguyên a nào không chia hết cho n ta có ≡ 1 (mod n).

Ưu điểm: Hiệu suất cao, thuật toán xử lý nhanh với các số nguyên lớn.

Hạn chế: Xác suất sai sót cao, dễ bị lừa bởi các số Carmichael.

**1.2.9.2 Thuật toán Miller- Rabin**

Để giải quyết hạn chế của hai thuật toán trên, thuật toán Miller-Rabin là một trong những lựa chọn phổ biến và hiệu quả.

Mô tả: Thuật toán này dựa trên định lý Fermat nhỏ và mở rộng để kiểm tra xem số nguyên n có là số nguyên tố hay không bằng cách: Xét số lẻ 𝑛 > 1 − Phân tích (𝑛 – 1) thành dạng 2𝑠 ∗ 𝑑 (với 𝑠, 𝑑 là số nguyên dương, 𝑑 là số lẻ)

Xét mỗi số nguyên 𝑎 trong khoảng [2, min (𝑛 − 1, ⌊⌋

Nếu ≢ 1(𝑚𝑜𝑑 𝑛) và ≢ −1(𝑚𝑜𝑑 𝑛) với mọi 𝑟 từ 0 đến (𝑠 – 1) thì 𝑛 không phải là số nguyên tố.

Nếu n vượt qua được tất cả các lần thử với a ở trên thì 𝑛 là số nguyên tố

Cách lựa chọn số lần kiểm tra k: Số lần kiểm tra k trong thuật toán Miller Rabin được chọn dựa trên độ chính xác mong muốn. Thường người ta chọn k như sau:

Nếu muốn mức độ chính xác cao (xác suất sai lầm < 2^-100), thì chọn k = 50.

Nếu muốn mức độ chính xác trung bình (xác suất sai lầm < 2^-50), thì chọn k = 25.

Nếu muốn mức độ chính xác cơ bản (xác suất sai lầm < 2^-20), thì chọn k = 10.

Cách chọn k như trên dựa trên các lý do sau:

Xác suất sai lầm của thuật toán Miller-Rabin khi kiểm tra số nguyên tố

n với k lần kiểm tra là tối đa

Khi k = 50, xác suất sai lầm sẽ nhỏ hơn 2−100, tức là khoảng 1 trên 100 triệu tỷ, đây là mức độ chính xác rất cao.

Khi k = 25, xác suất sai lầm sẽ nhỏ hơn 2−50, tức là khoảng 1 trên 1 triệu tỷ, đây là mức độ chính xác trung bình.

Khi k = 10, xác suất sai lầm sẽ nhỏ hơn 2−20, tức là khoảng 1 trên 1 triệu, đây là mức độ chính xác cơ bản.

Việc lựa chọn k phụ thuộc vào mục đích sử dụng và độ tin cậy cần thiết. Trong thực tế, k = 25 thường được sử dụng như một giá trị cân bằng giữa độ chính xác và hiệu suất.

Kiến thức cơ sở về ngôn ngữ lập trình C++ :

Ngôn ngữ C++ được Bjarne Stroustrup phát triển từ ngôn ngữ C từ cuối thập niên 1970.

C++ là một phiên bản mở rộng của ngôn ngữ C, kết hợp tất cả các tính năng đã có của C.

C++ được coi như là ngôn ngữ bậc trung (middle-level), kết hợp các đặc điểm và tính năng của ngôn ngữ bậc cao và bậc thấp.

C++ có thể dùng để lập trình nhúng, lập trình hệ thống, hoặc những ứng dụng, game…

C++ là ngôn ngữ "đa hướng". Nghĩa là nó hướng cấu trúc giống C và có thêm một tính năng cực kỳ quan trọng đó là tính năng hướng đối tượng. Các bạn sẽ được học phần hướng đối tượng của C++ trong serial Lập trình hướng đối tượng C++.

C++ là một trong những ngôn ngữ lập trình phổ biến trên thế giới.

C++ là ngôn ngữ tiềm năng với những điểm mạnh nổi bật như:

Tính phổ biến: Lập trình viên C++ là một trong những công việc được các nhà [tuyển dụng](https://careerviet.vn/viec-lam/tat-ca-viec-lam-vi.html) trên toàn cầu tìm kiếm nhiều. Điều này có thể cho thấy C++ là một trong những [ngôn ngữ lập trình phổ biến nhất trên thế giới](https://vieclamit.careerviet.vn/advices/top-15-ngon-ngu-lap-trinh-pho-bien-nhat-the-gioi-ban-da-biet-het-chua.35A4EDBC.html).

Dễ hiểu và dễ sử dụng: C++ được biết đến là ngôn ngữ lập trình đơn giản nhất. Lập trình này hoạt động theo nguyên tắc phân nhỏ các nhiệm vụ lớn thành các nhiệm vụ nhỏ cho bất kỳ dự án nào. Cấu trúc câu lệnh dễ hiểu, ít từ khóa nên khá dễ học và dễ sử dụng.

Thư viện đa dạng, phong phú: có nhiều tài nguyên hỗ trợ cho ngôn ngữ này như đồ họa API, 2D, 3D, vật lý các thiết bị âm… giúp cho các lập trình viên dễ thực thi.

Đa mô hình: C++ cũng cho phép bạn lập trình theo cấu trúc tuyến tính, hướng chức năng, hướng đối tượng đa dạng tùy theo yêu cầu của người lập trình.

Tính di động cao: một đoạn mã code C++ có thể chuyển đổi dễ dàng khi chạy trong macOS, Window hay Linux. Nhờ đó mà lập trình viên không phải tốn thời gian viết lại chương trình khi sử dụng nó để chạy trên nhiều platform khác nhau.

Có nhiều công cụ, phần mềm và IDE hỗ trợ giúp bạn đơn giản hóa công việc.

Ngôn ngữ lập trình C++ cơ bản là nền tảng định hướng cho việc tiếp cận sau này với các ngôn ngữ lập trình khác.

Thư viện C++

Thư viện chuẩn C++ dùng lại thư viện chuẩn C với một số điều chỉnh nhỏ để giúp nó hoạt động tốt hơn với ngôn ngữ C++. Một bộ phận lớn khác của thư viện C++ dựa trên [Thư viện tiêu bản chuẩn](https://vi.wikipedia.org/w/index.php?title=Th%C6%B0_vi%E1%BB%87n_ti%C3%AAu_b%E1%BA%A3n_chu%E1%BA%A9n&action=edit&redlink=1) (hay còn gọi là STL - viết tắt từ chữ Standard Template Library). Thư viện này có nhiều công cụ hữu dụng như là các [thùng chứa](https://vi.wikipedia.org/w/index.php?title=Th%C3%B9ng_ch%E1%BB%A9a_(c%E1%BA%A5u_tr%C3%BAc_d%E1%BB%AF_li%E1%BB%87u)&action=edit&redlink=1) (ví dụ như [vector](https://vi.wikipedia.org/wiki/Vect%C6%A1), [danh sách liên kết](https://vi.wikipedia.org/wiki/Danh_s%C3%A1ch_li%C3%AAn_k%E1%BA%BFt) và [biến lặp](https://vi.wikipedia.org/w/index.php?title=Bi%E1%BA%BFn_l%E1%BA%B7p&action=edit&redlink=1) (tổng quát hóa từ khái niệm [con trỏ](https://vi.wikipedia.org/wiki/Con_tr%E1%BB%8F)) để cung cấp những thùng chứa này sự truy cập giống như là truy cập mảng. Xa hơn nữa, bảng (đa) [ánh xạ](https://vi.wikipedia.org/wiki/%C3%81nh_x%E1%BA%A1) ([mảng kết hợp](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%A3ng_k%E1%BA%BFt_h%E1%BB%A3p&action=edit&redlink=1)) và (đa) tập, tất cả được cung cấp để có thể xuất ra các [giao diện](https://vi.wikipedia.org/wiki/Giao_di%E1%BB%87n) tương thích. Do đó, có thể dùng tiêu bản để viết các thuật toán tổng quát mà chúng làm việc được với bất kì thùng chứa nào hay với bất kì dãy nào được định nghĩa bởi biến lặp. Giống như C, các tính năng của thư viện này thì được truy cập bởi việc sử dụng [lệnh dẫn hướng](https://vi.wikipedia.org/w/index.php?title=L%E1%BB%87nh_d%E1%BA%ABn_h%C6%B0%E1%BB%9Bng&action=edit&redlink=1) #include để bao gồm một tập tin [tiêu đề chuẩn](https://vi.wikipedia.org/w/index.php?title=Ti%C3%AAu_%C4%91%E1%BB%81_chu%E1%BA%A9n&action=edit&redlink=1). C++ cung ứng 69 tiêu đề chuẩn, trong đó có 19 tiêu đề không còn hiệu lực nữa.

Vì thư viện chuẩn được thiết kế bởi những chuyện gia hàng đầu và đã được chứng minh trong toàn bộ lịch sử kỹ nghệ, các thành phần của thư viện này được khuyến cáo sử dụng thay vì dùng những phần viết tay bên ngoài hay những phương tiện cấp thấp khác. Ví dụ, dùng std:vector hay std::string thay vì dùng kiểu mảng đơn thuần sẽ không những là cho "đời sống dễ thở hơn", mà còn là một cách hữu hiệu để viết phần mềm được an toàn và linh hoạt hơn.

STL nguyên là một thư viện của hãng [HP](https://vi.wikipedia.org/wiki/Hewlett-Packard) và sau đó là của [SGI](https://vi.wikipedia.org/w/index.php?title=SGI&action=edit&redlink=1), trước khi nó được nhận vào thành chuẩn C++. Tiêu chuẩn thì không tham chiếu nó bằng cái tên "STL", khi đa phần nó chỉ là bộ phận tiêu chuẩn. Tuy vậy, nhiều người vẫn dùng khái niệm "STL" này để phân biệt nó với phần còn lại của thư viện C++ như là IOstream, quốc tế hóa (ký tự và ngôn ngữ trình bày), chẩn đoán, thư viện C,...

Các Thư Viện Phổ Biến

<bitset> -- Định nghĩa lớp tiêu bản bitset và các hàm hỗ trợ cho việc hiển thị và điều chỉnh các dãy bit có độ dài cố định.

<deque> -- Định nghĩa lớp tiêu bản deque (hàng) và nhiều tiêu bản hỗ trợ.

<list> -- Định nghĩa lớp tiêu bản list (danh sách) và nhiều tiêu bản hỗ trợ.

<map> -- Định nghĩa lớp tiêu bản map và <multimap> và nhiều tiêu bản hỗ trợ.

<queue> -- Định nghĩa các lớp tiêu bản priority\_queue và queue cùng với nhiều tiêu bản hỗ trợ.

<set> -- Định nghĩa lớp tiêu bản set và nhiều tiêu bản hỗ trợ.

<stack> -- Định nghĩa lớp tiêu bản stack (chồng) và các tiêu bản hỗ trợ.

<vector> -- Định nghĩa lớp tiêu bản vector và nhiều tiêu bản hỗ trợ.

Tổng quát

<algorithm> -- Định nghĩa nhiều hàm thực thi các thuật toán về thùng chứa trong Thư viện tiêu bản chuẩn STL.

<functional> -- Định nghĩa các hàm nhằm giúp cho việc xây dựng các đối tượng hàm trong Thư viện tiêu bản chuẩn (còn có tên là hàm tử).

<iterator> -- Định nghĩa các biến lặp cơ bản, định nghĩa sẵn các biến lặp, biến lặp dòng, cũng như là các tiêu bản hỗ trợ.

<locale> -- Định nghĩa các lớp tiêu bản và các hàm mà chương trình C++ có thể dùng để tạo đặc tính đóng và điều chỉnh các yếu tố định dạng và biểu thị của các số, tiền tệ, dữ liệu về ngày tháng bao gồm hỗ trợ ngôn ngữ quốc tế cho sự phân lớp ký tự và sự sắp xếp dãy ký tự.

<memory> -- Định nghĩa một lớp, một toán tử và nhiều tiêu bản nhằm giúp cấp phát và trả về (vùng nhớ) của các đối tượng.

<stdexcept> -- Định nghĩa nhiều lớp chuẩn dùng cho việc báo cáo các ngoại lệ. Các lớp tạo nên một hệ phân nhánh mà tất cả đều dẫn xuất từ lớp exception và bao gồm hai kiểu ngoại lệ thông dụng: lỗi lô-gíc và lỗi trong thời gian thi hành.

<utility> -- Định nghĩa các kiểu, hàm, và toán tử trong Thư viện tiêu bản chuẩn STL nhằm giúp cho việc xây dựng và quản lý các cặp đối tượng. Nó hữu dụng, khi hai đối tượng cần được hành xử như chúng là một.

Các dãy ký tự

<string> -- Định nghĩa thùng chứa lớp tiêu bản basic\_string và nhiều hỗ trợ cho các tiêu bản.khung

Streams và Input/Output

<fstream> -- Định nghĩa các lớp hỗ trợ các phép toán cho dòng xuất nhập (iostream) trên các chuỗi chứa trong các tập tin bên ngoài.

<ios> -- Định nghĩa nhiều kiểu và hàm cơ bản cho các xử lý của các dòng xuất nhập. Tiêu đề này thường bao gồm nhiều tiêu đề dòng xuất nhập khác mà ít khi được khai báo qua lệnh #include trực tiếp.

<iostream> -- Định nghĩa các đối tượng để đọc và viết ra các dòng tiêu chuẩn (standard stream). Đây thường là tiêu đề duy nhất cần phải bao gồm để tiến hành xuất và nhập (dữ liệu) từ C++.

<iosfwd> -- Khai báo trước các tham chiếu cho nhiều tiêu bản để dùng suốt quá trình xuất nhập dòng. Tất cả các tiêu bản như vậy được định nghĩa trong các tập tin tiêu đề chuẩn khác. Chỉ sử dụng tập tin tiêu đề này khi cần có một trong số các khai báo của nó, nhưng không phải là định nghĩa của nó.

<iomanip> -- Định nghĩa nhiều phép điều chỉnh mà nó lấy vào một đối số.

<istream> -- Định nghĩa lớp tiêu bản basic\_istream, tiêu bản này làm trung gian cho các sự chèn vào và lấy ra (dữ liệu). Tiêu đề cũng định nghĩa một bộ điều chỉnh. Tập tin tiêu đề này thường được sử dụng bởi các tiêu đề về dòng xuất nhập khác, nó hiếm khi được bao gồm trực tiếp trong mã nguồn của người lập trình.

<ostream> -- Định nghĩa lớp tiêu bản basic\_ostream, tiêu bản này làm trung gian cho các sự chèn (dữ liệu) vào cho các dòng xuất nhập. Tiêu đề cũng định nghĩa một bộ điều chỉnh. Tập tin tiêu đề này thường được sử dụng bởi các tiêu đề về dòng xuất nhập khác, nó hiếm khi được bao gồm trực tiếp trong mã nguồn của người lập trình.

<sstream> -- Định nghĩa nhiều lớp tiêu bản mà nó hỗ trợ các phép toán của dòng xuất nhập trên các chuỗi (dữ liệu) chứa trong một đối tượng mảng đã được cấp phát vùng nhớ. Những chuỗi này rất dễ chuyển đổi để tạo thành các đối tượng của lớp tiêu bản basic\_string.

<streambuf> -- Bao gồm tiêu đề chuẩn cho các dòng xuất nhập của <streambuf> dùng để định nghĩa lớp tiêu bản basic\_streambuf. Lớp này là cơ sở cho phép toán của các lớp của dòng xuất nhập. Tiêu đề này thường được bao gồm sẵn trong các tiêu bản về dòng xuất nhập khác, nó hiếm khi được bao gồm trực tiếp trong mã nguồn của người lập trình.

Các số

<complex> -- Định nghĩa lớp tiêu bản complex và hỗ trợ nhiều tiêu bản.

<numeric> -- Định nghĩa các hàm tiêu bản thùng chứa nhằm tiến hành các thuật toán cung cấp cho việc xử lý số.

<valarray> -- Định nghĩa lớp tiêu bản valarray hỗ trợ các lớp tiêu bản và các hàm.

Hỗ trợ ngôn ngữ lập trình

<exception> --Định nghĩa nhiều kiểu và hàm liên hệ tới việc xử lý các ngoại lệ. Việc xử lý ngoại lệ được dùng trong tình huống mà máy tính có thể phục hồi từ các lỗi. Nó cung ứng ý nghĩa cho việc trả về quyền điều khiển cho chương trình.

<limits> --Định nghĩa các lớp tiêu bản numeric\_limits và lo về việc biểu thị thứ tự của các điểm động và việc làm tròn.

<new> -- Định nghĩa nhiều kiểu và hàm mà nó điều khiển việc cấp phát cũng như trả về (bộ nhớ) của kho dự trữ dưới sự kiểm soát của chương trình. Nó cũng định nghĩa các thành phần cho việc báo cáo về quản lý của kho dự trữ.

<typeinfo> -- Hỗ trợ cho nhận dạng kiểu động.

**1.3. Nội dung nghiên cứu**

Lý do nghiên cứu về an ninh mạng:

Giảm thiểu mối đe dọa mạng: Sự gia tăng các cuộc tấn công mạng, như phishing, ransomware, DDoS, khiến việc giảm thiểu và ngăn chặn chúng trở nên cấp thiết.

Bảo vệ dữ liệu: Đảm bảo dữ liệu cá nhân, doanh nghiệp và quốc gia không bị mất mát hay đánh cắp.

Hỗ trợ kinh tế và xã hội: Đảm bảo an toàn cho giao dịch trực tuyến, hệ thống tài chính và cơ sở hạ tầng quan trọng.

Quy định pháp lý: Đáp ứng các yêu cầu tuân thủ về bảo mật thông tin trong nhiều ngành công nghiệp.

Phát triển công nghệ: Đáp ứng nhu cầu về bảo mật ngày càng cao trong quá trình chuyển đổi số.

Nội dung nghiên cứu chính về an ninh mạng

Phát hiện và phòng ngừa tấn công mạng:

Các biện pháp phòng thủ chủ động như tường lửa, mã hóa, và phát hiện xâm nhập.

Phân tích hành vi của mã độc và các lỗ hổng phần mềm.

Bảo vệ dữ liệu và quyền riêng tư:

Các công nghệ mã hóa nâng cao (AES, RSA).

Quản lý quyền truy cập và xác thực (MFA, sinh trắc học).

Trí tuệ nhân tạo trong an ninh mạng:

Ứng dụng AI để phát hiện tấn công, phân tích mẫu và dự đoán nguy cơ.

Phát triển hạ tầng an ninh:

An toàn cho các hệ thống IoT, điện toán đám mây, và mạng 5G.

Kiến trúc Zero Trust (Không tin tưởng tuyệt đối).

Phân tích pháp y số (Digital Forensics):

Điều tra, thu thập chứng cứ, và truy vết tội phạm mạng.

Đào tạo và nhận thức:

Nâng cao nhận thức cộng đồng về an ninh mạng.

Xây dựng đội ngũ chuyên gia bảo mật chuyên sâu.

Chính sách và hợp tác quốc tế:

Xây dựng quy định pháp luật và chiến lược quốc gia về an ninh mạng.

Tăng cường hợp tác quốc tế để đối phó với các mối đe dọa toàn cầu.

⇒ Nghiên cứu an ninh mạng không chỉ bảo vệ các hệ thống hiện tại mà còn tạo nền tảng vững chắc cho sự phát triển bền vững của xã hội số.

# **Chương II: Kết quả nghiên cứu**

## **2.1. Nghiên cứu tìm hiểu hệ mã hóa khóa công khai**

### **2.1.1. Khái niệm hệ mã hoá công khai**

Khái niệm tổng quan:

Hệ mã hóa khóa công khai (Public-Key Cryptography), hay còn gọi là mã hóa bất đối xứng, là một phương pháp mã hóa trong đó sử dụng hai khóa khác biệt nhưng có mối quan hệ toán học chặt chẽ: khóa công khai (public key) và khóa bí mật (private key).

Khóa công khai: Được chia sẻ công khai, cho phép bất kỳ ai có khóa này đều có thể mã hóa thông điệp hoặc kiểm tra chữ ký số.

Khóa bí mật: Chỉ được lưu trữ và bảo mật bởi người nhận, cho phép giải mã thông điệp đã được mã hóa hoặc tạo ra chữ ký số.

Hệ mã hóa khóa công khai dựa trên nguyên tắc sau:

Tạo khoá:

Một cặp khóa bao gồm khóa công khai (có thể chia sẻ với bất kỳ ai) và khóa bí mật (chỉ người sở hữu giữ).

Hai khóa này được tạo ra bằng các thuật toán mật mã dựa trên các bài toán toán học khó giải (ví dụ: phân tích số nguyên tố lớn hoặc bài toán logarit rời rạc).

Quá trình mã hóa:

Người gửi sử dụng khóa công khai của người nhận để mã hóa dữ liệu.

Do chỉ khóa bí mật tương ứng mới có thể giải mã dữ liệu này, điều đó đảm bảo rằng chỉ người sở hữu khóa bí mật mới có thể đọc được thông tin.

Quá trình giải mã: Người nhận sử dụng khóa bí mật để giải mã dữ liệu và khôi phục lại thông điệp ban đầu.

### **2.1.2. Đặc điểm của hệ mã hóa khóa công khai**

#### **2.1.2.1. Tính bất đối xứng**

Khóa công khai và khóa bí mật là hai khóa riêng biệt nhưng có mối quan hệ chặt chẽ. Một khóa được dùng để mã hóa hoặc giải mã, và khóa còn lại sẽ thực hiện thao tác ngược lại. Khi thông điệp được mã hóa bằng khóa công khai, chỉ khóa bí mật tương ứng mới có thể giải mã, bảo vệ thông tin ngay cả khi khóa công khai bị lộ.

#### **2.1.2.2. Tính bảo mật dựa trên toán học**

Hệ mã hóa khóa công khai dựa vào các bài toán toán học khó giải, như phân tích số nguyên tố trong RSA hoặc bài toán logarit rời rạc trong ECC. Những bài toán này yêu cầu tài nguyên tính toán lớn và khiến việc suy luận khóa bí mật trở nên bất khả thi, đảm bảo an toàn trước các phương thức tấn công hiện đại.

#### **2.1.2.3. Không cần kênh an toàn để phân phối khóa**

Khác với mã hóa khóa đối xứng, hệ mã hóa khóa công khai không yêu cầu chia sẻ khóa bí mật qua kênh an toàn. Người gửi chỉ cần khóa công khai của người nhận để mã hóa thông điệp, và khóa này có thể được chia sẻ công khai qua bất kỳ kênh nào, kể cả không an toàn, giảm thiểu nguy cơ bị đánh cắp khóa bí mật.

#### **2.1.2.4. Tính linh hoạt**

Hệ mã hóa khóa công khai không chỉ dùng để mã hóa dữ liệu mà còn có các ứng dụng khác như xác thực danh tính (chữ ký số), đảm bảo tính toàn vẹn tài liệu, và trao đổi khóa bí mật cho giao tiếp lâu dài. Điều này kết hợp ưu điểm của mã hóa đối xứng và khóa công khai, mang lại bảo mật và hiệu suất cao.

### **2.1.3. Cơ chế Hoạt động**

Hệ mã hóa khóa công khai hoạt động qua ba bước chính:

Bước 1: Tạo cặp khóa

Một cặp khóa gồm khóa công khai và khóa bí mật được tạo ra bởi các thuật toán mã hóa, ví dụ RSA hoặc ECC.

Quá trình này thường sử dụng các số nguyên tố rất lớn để đảm bảo tính bảo mật.

Bước 2: Mã hóa dữ liệu

Người gửi sử dụng khóa công khai của người nhận để mã hóa thông điệp. Quá trình mã hóa chuyển đổi thông điệp gốc thành dạng mã hóa mà chỉ người nhận mới có thể giải mã.

Bước 3: Giải mã dữ liệu

Người nhận sử dụng khóa bí mật của mình để giải mã thông điệp mã hóa và khôi phục lại thông tin gốc.

### **2.1.4. Các Thuật Toán Mã hóa Khóa Công khai Phổ biến**

#### **2.1.4.1. RSA (Rivest-Shamir-Adleman)**

RSA là một trong những thuật toán mã hóa khóa công khai đầu tiên và phổ biến nhất. Nguyên lý hoạt động dựa trên độ khó của việc phân tích một số nguyên lớn thành thừa số nguyên tố. Trong RSA, mỗi người dùng tạo ra một cặp khóa gồm khóa công khai và khóa bí mật, liên quan chặt chẽ qua các tính toán toán học.

Ứng dụng: Sử dụng trong mã hóa dữ liệu qua Internet (HTTPS, TLS) và tạo chữ ký số để đảm bảo tính toàn vẹn và xác thực nguồn gốc tài liệu.

#### **2.1.4.2. ECC (Elliptic Curve Cryptography)**

ECC là một thuật toán mã hóa dựa trên các tính chất toán học của đường cong elliptic. Trong hệ này, mỗi khóa công khai tương ứng với một điểm trên đường cong, và khóa bí mật là một số nguyên ngẫu nhiên. ECC cung cấp cùng mức độ bảo mật với độ dài khóa ngắn hơn so với RSA, giúp tăng hiệu suất và giảm chi phí tính toán.

Ứng dụng: Mã hóa dữ liệu trong các thiết bị hạn chế tài nguyên như IoT và sử dụng trong giao thức bảo mật TLS, cùng với chữ ký số (ECDSA).

#### **2.1.4.3. Diffie-Hellman**

Diffie-Hellman là một thuật toán trao đổi khóa, cho phép hai bên thiết lập một khóa bí mật chung thông qua một kênh không an toàn. Thuật toán này dựa trên bài toán logarit rời rạc, đảm bảo rằng khóa bí mật chỉ có thể được tính toán bởi hai bên tham gia trao đổi.

Ứng dụng: Thiết lập khóa bí mật chung trong các giao thức bảo mật như IPsec và TLS, giúp bảo vệ thông tin trong các kết nối mạng.

#### **2.1.4.4. DSS (Digital Signature Standard)**

DSS là một tiêu chuẩn chữ ký số được phát triển bởi NIST, dựa trên thuật toán DSA (Digital Signature Algorithm). DSS không sử dụng để mã hóa mà chỉ tập trung vào việc tạo và xác thực chữ ký số nhằm đảm bảo tính toàn vẹn và nguồn gốc của tài liệu.

Ứng dụng: Xác thực tài liệu trong các giao dịch điện tử, sử dụng rộng rãi trong hệ thống chính phủ và ngân hàng để đảm bảo tính toàn vẹn và nguồn gốc.

### **2.1.5. Ưu Điểm và Hạn Chế**

#### **2.1.5.1. Ưu điểm**

Tăng cường bảo mật: Hệ mã hóa khóa công khai không cần chia sẻ khóa bí mật qua kênh truyền tải, giúp giảm nguy cơ bị đánh cắp thông tin, đặc biệt trong các môi trường mạng không an toàn như Internet công cộng.

Đa năng: Hỗ trợ mã hóa dữ liệu, xác thực danh tính thông qua chữ ký số và bảo đảm tính toàn vẹn của thông tin; đồng thời còn được dùng để trao đổi khóa trong các hệ thống mã hóa khóa đối xứng, kết hợp tính bảo mật và hiệu suất.

Tương thích tốt với các hệ thống hiện đại: Tích hợp dễ dàng với các giao thức bảo mật như HTTPS và TLS, cũng như các ứng dụng tiên tiến như blockchain và hệ thống VPN.

#### **2.1.5.2. Hạn chế**

Hiệu suất thấp hơn mã hóa khóa đối xứng: Các thuật toán trong hệ mã hóa khóa công khai đòi hỏi nhiều phép toán phức tạp như lũy thừa modulo hoặc tính toán trên đường cong elliptic, khiến tốc độ mã hóa và giải mã chậm hơn, đặc biệt với lượng dữ liệu lớn.

Dễ bị tấn công nếu khóa ngắn: Việc sử dụng khóa ngắn sẽ giảm độ bảo mật, tạo điều kiện cho các phương pháp tấn công brute force hoặc phân tích ngược phá vỡ hệ thống, đặc biệt khi máy tính lượng tử trở nên phổ biến.

### **2.1.6. Ứng Dụng Thực Tiễn**

Giao thức HTTPS: Sử dụng để mã hóa các kết nối giữa trình duyệt và máy chủ web, giúp bảo vệ thông tin người dùng như mật khẩu hoặc dữ liệu giao dịch khỏi bị đánh cắp trong quá trình truyền tải.

Chữ ký số: Được dùng để xác thực nguồn gốc tài liệu và đảm bảo tính toàn vẹn, giúp ngăn chặn việc chỉnh sửa hoặc giả mạo thông tin trong các giao dịch điện tử hoặc tài liệu pháp lý.

Blockchain: Sử dụng để bảo vệ các giao dịch và đảm bảo danh tính người tham gia, bằng cách tạo và xác minh chữ ký số cho các giao dịch trong các hệ thống tiền điện tử và hợp đồng thông minh.

Email bảo mật: Dùng để mã hóa nội dung email, đảm bảo chỉ người nhận mới có thể đọc được, đồng thời xác thực rằng email được gửi từ đúng người.

Trao đổi khóa trong VPN: Được sử dụng để thiết lập khóa chung giữa hai bên trong kết nối mạng riêng ảo, bảo vệ dữ liệu khỏi bị truy cập trái phép qua mạng công cộng.