ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

ĐINH THỊ THÚY

NGHIÊN CỨU VÀ PHÁT TRIỂN ỨNG DỤNG JAVACARD

Ngành: Công nghệ thông tin
Chuyên ngành: Quản lý Hệ thống thông tin
Mã số: Chuyên ngành đào tạo thí điểm

TÓM TẮT LUẬN VĂN THẠC SĨ CÔNG NGHỆ THÔNG TIN

HÀ NỘI – 2017

Người hướng dẫn khoa học: **TS. Lê Phê Đô – TS. Phùng Văn Ổn**

Phản biện 1:PGS.TS. Nguyễn Trí Thành

Phản biện 2:TS. Nguyễn Ngọc Cương

Luận văn được bảo vệ trước Hội đồng chấm luận văn thạc sĩ
tại trường Đại học Công nghệ
Vào hồi: ngày11 tháng 08 năm 2017

Có thể tìm hiểu luận văn tại:
Trung tâm thông tin thư viện Đại học Quốc gia Hà Nội

MỞ ĐẦU

1. Tính cấp thiết của đề tài luận văn

Ngày nay, sự hội nhập kinh tế sâu rộng đã mang đến cho người tiêu dùng Việt Nam cơ hội tiếp cận với những xu hướng hiện đại của thế giới. Con người dần chuyển sang sử dụng các dịch vụ thông minh hơn, tiện lợi hơn để đáp ứng các nhu cầu cuộc sống một cách hiện đại, tối ưu. Giờ đây người tiêu dùng có thể dễ dàng mua sắm, thanh toán các dịch vụ sinh hoạt, giao thông, y tế mà không cần phải mất thời gian và công sức tới các điểm giao dịch như trước thay vào đó là việc sử dụng một thiết bị đơn giản nhỏ gọn là thẻ thông minh. Sự phát triển nhanh chóng của công nghệ bán dẫn cho phép các nhà sản xuất chip tạo ra những con chip hay thẻ thông minh ngày càng nhỏ gọn cùng với sức mạnh tính toán cao. Tuy nhiên việc có quá nhiều nhà sản xuất chip, công việc phát triển ứng dụng cho thẻ thông minh gặp khó khăn về sự tương thích. Do đó nhu cầu về một nền tảng chung bên trong chip được đặt ra, công nghệ Java Card được phát triển để phục vụ mục đích này. Với việc tạo ra một môi trường ảo chung trên tất cả các hệ điều hành hỗ trợ JavaCard, công nghệ này đã giúp cho việc phát triển ứng dụng chip trở nên dễ dàng giúp tiết kiệm thời gian nghiên cứu phát triển.

Hình thức mua sắm trực tuyến đang ngày càng phổ biến và người tiêu dùng sẽ dễ dàng chọn lựa, sở hữu những món hàng yêu thích hay săn tìm các chương trình giảm giá, khuyến mãi hấp dẫn khi sở hữu thẻ tín dụng. Thẻ tín dụng là phương tiện thanh toán phù hợp với lối sống hiện đại. Tuy nhiên, quy trình đăng ký thẻ tín dụng mất khá nhiều thời gian, người tiêu dùng sau khi chuẩn bị giấy tờ, tới chi nhánh ngân hàng để đăng ký, thời gian đăng ký hạn chế trong giờ hành chính gây bất tiện cho người đăng ký thẻ mới. Ngoài ra thời gian chờ đợi thẻ cũng mất từ năm đến bẩy ngày và phải lên đúng chi nhánh nơi mình đã đăng ký để nhận thẻ.

Đi đôi với việc phổ dụng các giao dịch thông qua mạng Internet dẫn đến nguy cơ mất an toàn thông tin khi sử dụng thẻ tín dụng. Do đó, vấn đề đặt ra là làm thế nào đảm bảo an toàn thông tin trong giao dịch trực tuyến và đăng ký thẻ. Chúng ta cần có các giải pháp đảm bảo an toàn thông tin sử dụng được xây dựng dựa trên lý thuyết mật mã, an toàn bảo mật thông tin. Các nhà khoa học đã phát minh ra những hệ mật mã như RSA, Elgamal, SHA1, SHA2, SHA3…nhằm che dấu thông tin cũng như là làm rõ chúng để tránh sự nhòm ngó của những kẻ cố tình phá hoại. Mặc dù rất an toàn nhưng có độ dài khoá lớn nên trong một số lĩnh vực không thể ứng dụng được. Chính vì vậy hệ mật trên đường cong elliptic ra đời. Đây là hệ mật được đánh giá là hệ mật có độ bảo mật an toàn cao và hiệu quả hơn nhiều so với hệ mật công khai khác.

Ở phạm vi đề tài này tôi đặt ra vấn đề nghiên cứu ứng dụng hệ mật trên đường cong Elliptic vào bảo mật thẻ thông minh nhằm đảm bảo an toàn thông tin trong việc đăng ký thẻ trực tuyến cũng như giao dịch trực tuyến trên Internet.

Mục đích nghiên cứu:

Luận văn đề cập đến công việc thực tiễn hiện nay là việc phát triển ứng dụng cho các loại thẻ thông minh hỗ trợ công nghệ Java Card. Phần lý thuyết trình bày các kiến thức liên quan về thẻ thông minh, công nghệ Java Card, cung cấp nền tảng cơ sở cho lập trình viên trước khi xây dựng ứng dụng hay thiết kế hệ thống sử dụng công nghệ Java Card. Phần thực nghiệm sử dụng cơ sở lý thuyết ở trên để cải tiến quy trình đăng ký thẻ tín dụng bằng cách áp dụng chữ ký số trên hệ mật đường cong Elliptic vào việc đăng ký thẻ tín dụng trực tuyến nhằm bảo đảm an toàn thông tin trong thẻ tín dụng.

2. Nội dung của đề tài, các vấn đề cần giải quyết:

a. Hướng nghiên cứu:

- Công nghệ Java card
- Ứng dụng mật mã đường cong Elliptic trong bảo mật thẻ thông minh.

b. Ngoài phần mở đầu, kết luận, nội dung luận văn gồm những chương sau:

Chương 1: Tổng quan thẻ thông minh.

Chương 2: Công nghệ Java Card.

Chương 3: Mật mã đường cong Elliptic.

Chương 4: Ứng dụng hệ mật đường cong elliptic trong bảo mật thẻ thông minh.

CHƯƠNG 1 TỔNG QUAN THẺ THÔNG MINH

Thẻ thông minh đang được ứng dụng rộng rãi tại Việt Nam trong nhiều lĩnh vực như viễn thông, ngân hàng, thương mại điện tử, điều khiển tự động, kiểm soát người và phương tiện… Các ứng dụng của thẻ thông minh rất thiết thực và tích hợp phần mềm điều khiển bởi ưu điểm vượt trội về khả năng lưu trữ, xử lý thông tin và bảo mật dữ liệu. Chương I trình bày cái nhìn tổng quan về thẻ thông minh.

1.1 Lịch sử phát triển thẻ thông minh

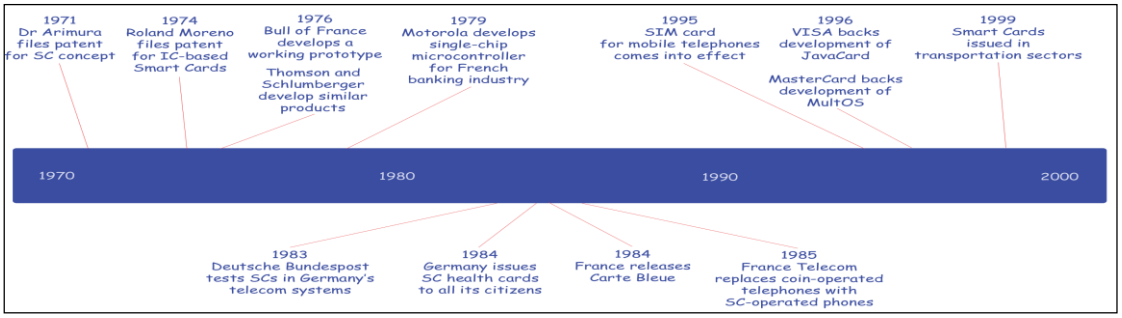
Có hai ý tưởng chính đã dẫn đến sự phát triển của thẻ thông minh. Ý tưởng đầu tiên xuất hiện bởi Tiến sĩ Kunitaka Arimura đến từ Nhật Bản, ông có thiết kế tích hợp dữ liệu lưu trữ và logic số học vào một miếng silicon, ông đã nộp bản quyền cho ý tưởng vào năm 1970. Ý tưởng thứ hai là kỹ sư người Đức Helmut Gröttrup và đồng nghiệp là Jürgen Dethloff, họ đã nộp bản quyền năm 1968[6]. Bằng sáng chế thẻ chip tự động này được công bố vào cuối năm 1982[6]. Năm 1974, Roland Moreno – một nhà phát minh của Pháp, đã gắn chip lên một tấm nhựa và cấp bằng sáng chế về thẻ nhớ và thiết bị đọc nó, được đặt tên là thẻ thông minh. Moreno đã thành lập công ty Innovatron để bán ý tưởng, Moreno được biết như là cha đẻ của mạch vi xử lý (Microchip).[6]

Hình 1.1 Chip tự động

Năm 1977, ba nhà sản xuất thương mại, Bull CP8, SGS Thomson và Schlumberger đã bắt đầu phát triển các sản phẩm của thẻ thông minh[6]. Năm 1978, Bull đăng ký bằng sáng chế về bộ vi xử lý một chip tự lập trình được (SPOM-self Programmable One-chip Microcomputer)[6].

Hình 1.2 Thẻ CP8

Đến năm 1981 những chiếc thẻ thông minh đã có nền tảng ở nhiều nước Tây Âu, một số các ngân hàng Châu Âu đồng ý thành lập một cơ quan quản lý mới cho phát triển thẻ, ứng dụng và tiêu chuẩn. Tổ chức này bao gồm các tổ chức tài chính từ Bỉ, Anh, Đan Mạch, Áo, Hà Lan, và liên minh cũ các ngân hàng Pháp.



Hình 1.3 Sơ đồ lịch sử phát triển thẻ thông minh

1.2 Cấu tạo và phân loại thẻ thông minh

Cấu tạo và phân loại thẻ thông minh:

Thẻ thông minh thông thường có kích thước cỡ một thẻ tín dụng và được làm bằng nhựa, thường là PVC (Polyvinyl chloride – là thẻ nhựa cứng và không có mùi) đôi khi là ABS (Acrylonnitrile, Butadiene, Styrene – là một thẻ nhựa chịu được sự va đập mạnh), thẻ có thể chứa một ảnh 3 chiều tránh lừa đảo. Kích thước theo chuẩn ID-1(ISO/IEC/7810) quy định là 85,60x53,98 mm hoặc chuẩn ID-000 kích thước 25x15 mm, có bề dày mặt thẻ là 0,76 mm.

Phân loại thẻ thông minh: Có hai cách phân loại thẻ thông minh dựa trên công nghệ chip hay phương thức đọc dữ liệu.

Phân loại dựa trên công nghệ chip: Theo công nghệ chip được chia làm hai loại là thẻ chip nhớ (memory chip) và thẻ chip vi xử lý (microprocessor chip) được gắn trên bề mặt thẻ.[5]

- *Thẻ chip nhớ* bao gồm hai thành phần chính là thẻ nhớ cho phép có thể truy cập, giao thức truyền thông. Ưu điểm của thẻ này là dễ sản xuất, dễ sử dụng, nhược điểm là hạn chế về bộ nhớ và tính bảo mật không cao.
- *Thẻ vi xử lý (microprocessor chip)* được cấu tạo bởi ba loại bộ nhớ, một bộ vi xử lý (CPU – Central Processing Unit), một bộ đồng xử lý mã hóa (Cryto coprocessor) và một giao diện thông tin (communication interface). Chức năng của CPU là điều khiển các bộ phận khác, xử lý thông tin và thực hiện các phép tính.

Phân loại dựa trên phương thức đọc dữ liệu: chia làm 3 loại là thẻ tiếp xúc, thẻ không tiếp xúc và thẻ lưỡng tính[5].

- *Thẻ tiếp xúc*: là loại thẻ có một diện tích tiếp xúc thường dễ nhận diện bởi có gắn con chip (màu vàng hoặc bạc) trên thân thẻ, tiếp điểm đó có diện tích khoảng 1cm^2, được chia thành các phần riêng biệt gồm đầu vào, đầu ra dữ liệu, tín hiệu reset (phục hồi trạng thái ban đầu của thẻ), tín hiệu xung đồng hồ, chân điện áp.

Hình 1.4 Thẻ thông minh tiếp xúc và đầu đọc thẻ.

- Thẻ không tiếp xúc

Hình 1.5 Thẻ không tiếp xúc

Hình 1.6 Thẻ thu phí giao thông và thẻ dùng cho việc giao thông công cộng.

- *Thẻ lưỡng tính*

Các ứng dụng tiêu biểu của thẻ thông minh:

✓ Định danh: Đối với các hệ thống cần xác nhận định danh được phép truy cập hệ thống như: Mạng viễn thông di động, tài khoản ngân hàng, chứng minh nhân dân điện tử, hộ chiếu điện tử hay hệ thống quản lý truy cập (Access Control) thì TTM được đại diện cho quyền truy cập các hệ thống này.

✓ Lưu trữ: Khả năng lưu trữ an toàn trên thẻ smartcard, cho phép lưu trữ những thông tin thuộc về chủ thẻ như thông tin y tế, thông tin cá nhân, chứng chỉ điện tử (thẻ bảo hiểm y tế, giấy phép lái xe điện tử, v.v...).

✓ Xác thực Offline: Ngoài các ứng dụng phổ biến nói trên, thẻ thông minh còn được dùng để kiểm tra tính xác thực thẻ thành viên không yêu cầu kết nối hệ thống trung tâm.

1.3 Ưu nhược điểm của thẻ thông minh

Ưu điểm của thẻ thông minh: Thẻ thông minh với cấu tạo chip có nhiều ưu điểm hơn so với các loại thẻ từ khác. Ưu điểm của thẻ thông minh là:

➢ *Thẻ thông minh được ứng dụng được trong nhiều lĩnh vực*

➢ *Tính bảo mật cao*

➢ *Khả năng lưu trữ thông tin lớn*

- *Khả năng xử lý thông tin nhanh*
- *Có nhiều dịch vụ hỗ trợ người dùng và đơn giản hóa thủ tục*
- *Sử dụng trên phạm vi quốc tế*

Hạn chế của thẻ thông minh

- *Dễ bị mất, dễ hư hỏng*
- *Vấn đề an toàn thẻ thông minh*
- *Tăng nguy cơ phạm tội*
- *Rủi ro về quyền riêng tư*
- *Rủi ro về việc phân phối thẻ thông minh*

1.4 Thách thức trong việc phát triển ứng dụng thẻ thông minh

Phát triển ứng dụng thẻ thông minh theo truyền thống là một quá trình dài và khó. Mặc dù các thẻ được chuẩn hóa về kích thước, hình dạng, và giao thức giao tiếp, các hoạt động bên trong khác nhau giữa các nhà sản xuất. Hầu hết các công cụ phát triển thẻ thông minh được xây dựng bởi các nhà sản xuất thẻ thông minh bằng cách sử dụng các công cụ ngôn ngữ lắp ráp chung và giả lập phần cứng chuyên dụng thu được từ các nhà cung cấp chip silicon. Hầu như không thể cho các bên thứ ba phát triển các ứng dụng một cách độc lập và bán chúng cho các tổ chức phát hành. Do đó, việc phát triển các ứng dụng thẻ thông minh đã được giới hạn trong một nhóm các chuyên gia giàu kinh nghiệm và chuyên viên lập trình, những người có kiến thức sâu rộng về phần cứng và phần mềm thẻ thông minh cụ thể.

Hơn nữa, các ứng dụng thẻ thông minh được phát triển để chạy trên nền tảng độc quyền, các ứng dụng từ các nhà cung cấp dịch vụ khác nhau không thể cùng tồn tại và cung cấp trên một thẻ duy nhất. Công nghệ Java Card là một giải pháp để vượt qua các trở ngại cản trở việc phát triển thẻ thông minh. Nó cho phép thẻ thông minh và thiết bị hạn chế bộ nhớ khác có thể chạy các ứng dụng (được gọi là applet) được viết bằng ngôn ngữ lập trình Java. Thông thường, công nghệ Java Card xác định nền tảng thẻ thông minh an toàn, di động và nhiều ứng dụng kết hợp nhiều lợi thế chính của ngôn ngữ Java [6].

1.5 Các hình thức tấn công trên thẻ thông minh

Thẻ thông minh có khả năng bảo mật cao bởi các thành phần vật lý của con chip đều ở dạng siêu nhỏ và chúng đều có khả năng chống lại những tấn công vật lý. Tuy nhiên mạnh mẽ là thế các thẻ thông minh vẫn có những yếu điểm các hacker luôn tìm thấy cảm hứng từ các biến chip nhỏ bé, các hacker đã phát triển một loạt các kỹ thuật để quan sát và ngăn chặn các hoạt động của thẻ thông minh để có thể tước đoạt quyền truy cập thông tin, lấy các thông tin hữu ích cũng như chiếm đoạt thông tin đó. Dưới đây sẽ mô tả các cuộc tấn công trên thẻ thông minh, hiện nay có ba cuộc tấn

công cơ bản: cuộc tấn công logic, cuộc tấn công phần cứng và cuộc tấn công kênh phụ (side - channel)[6].

Cuộc tấn công Logic: dựa vào những suy luận logic liên quan đên các thuật toán mã hóa hacker cố gắng khai thác lỗ hổng trong các lĩnh vực sau:

Triển khai phần mềm

Các lệnh ẩn

Định vị thông số và tràn bộ đệm

Giao thức mã hoá, thiết kế và cài đặ

Cuộc tấn công phần cứng: đòi hỏi các thiết bị hiện đại để có thể xâm nhập vào các vi mạch của thẻ (chip), hacker khai thác lỗi trong các lĩnh vực sau:

Tấn công xâm nhập

Tấn công nửa xâm nhập

Dung môi hóa học, chất tẩy, chất nhuộm

Kính hiển vi quang học và điện tử

Trạm thăm dò

Tấn công qua kênh phụ: Một cuộc tấn công kênh phụ cố gắng khai thác một số hiện tượng vật lý để phân tích hoặc sửa đổi hành vi của thẻ thông minh, chẳng hạn như thời gian thực hiện thao tác, năng lượng tiêu thụ điện, cường độ của điện trường vv…

Thời gian thực hiện thao tác

Năng lượng tiêu thụ điện

Cường độ của điện trường

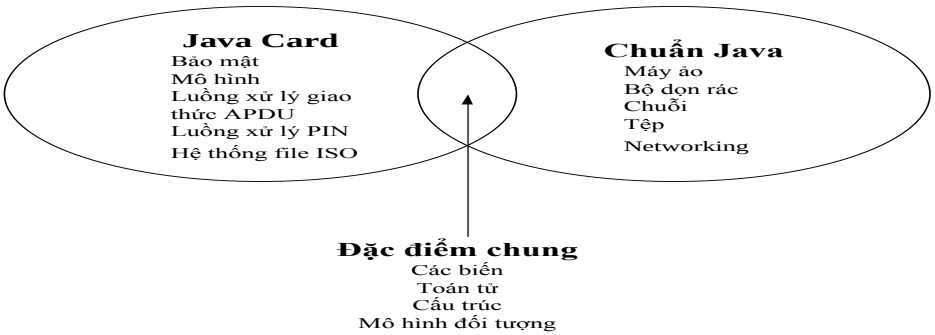
CHƯƠNG 2 CÔNG NGHỆ JAVACARD

Thẻ thông minh có nhiều ưu điểm nhưng cũng có thách thức để phát triển thẻ thông minh. Các ứng dụng thẻ thông minh được phát triển để chạy trên nền tảng độc quyền, các ứng dụng từ các nhà cung cấp dịch vụ khác nhau không thể cùng tồn tại và cung cấp trên một thẻ duy nhất. Đó là trở ngại cản trở việc phát triển ứng dụng trên thẻ thông minh. Công nghệ JavaCard được hình thành nhằm giải quyết vấn đề trên. Chương 2 đi sâu vào nền tảng công nghệ JavaCard.

2.1 Giới thiệu JavaCard

JavaCard là một công nghệ cho phép mang đến cho các trình ứng dụng Java applet có thể hoạt động một cách an toàn và bảo mật trên thẻ thông minh tương tự với các bộ nhớ nhỏ của các thiết bị lưu vết. Nó là nền tảng Java nhỏ nhất hướng tới các thiết bị nhúng. JavaCard là một phần nhỏ của Java được phát triển bởi Sun, được tích hợp bên trong các thiết bị, nó đơn giản hoá việc lập trình thẻ thông minh vì các tính năng hướng đối tượng của nó.

JavaCard có một số đặc điểm khác với Java thông thường. Về mặt ngôn ngữ lập trình, JavaCard là một phần thu gọn của Java, các cú pháp của Java, giống như tất cả các biến thể của Java và làm cho lập trình viên viết mã dễ dàng hơn vì không cần phải học một cú pháp nào khác[6].



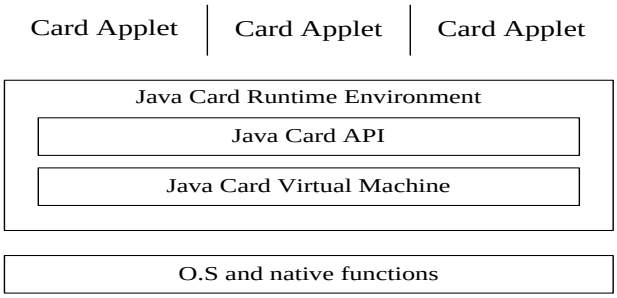
Hình 2.1 Các tính năng chung giữa Java Card và chuẩn Java

JavaCard phát triển với mục đích lưu trữ các thông tin nhạy cảm, công nghệ Java Card luôn đề cao tính bảo mật và đảm bảo điều này bằng các yếu tố khác nhau như: đóng gói dữ liệu, tường lửa ngăn cách ứng dụng, mã hóa dữ liệu, tạo ứng dụng dạng Applet. Mỗi ứng dụng khác nhau lưu trữ trên JavaCard đều được ngăn cách bởi một tường lửa để hạn chế và kiểm tra được sự truy cập dữ liệu từ ứng dụng này sang ứng dụng khác. Khả năng mã hóa của JavaCard cho phép dữ liệu được mã hóa bằng các dạng mã hóa thông dụng sử dụng khóa như mã hóa DES, 3DES, AES hay RSA.

2.2 Kiến trúc JavaCard

Về mặt ngôn ngữ lập trình thì JavaCard là một phần thu gọn của Java, công nghệ JavaCard cung cấp kiến trúc để phát triển ứng dụng mở cho thẻ thông minh và nó cũng được sử dụng để phát triển các ứng dụng cho các thiết bị có bộ nhớ rất nhỏ như SIM cho điện thoại di động, thẻ ATM gắn chip.

Dưới đây là mô hình kiến trúc [7]:



Hình 2.2 Kiến trúc tổng quát của công nghệ JavaCard

Công nghệ JavaCard bao gồm các bộ thông số kỹ thuật sau:

- JavaCard API: Một giao diện lập trình ứng dụng là lớp thư viện lõi của JavaCard.
- Máy ảo JavaCard: Mô tả các đặc tính của máy ảo để xử lý các ứng dụng của JavaCard
- JCRE: (Javacard runtime environment) mô tả hành vi chi tiết về thời gian chạy, chẳng hạn như cách bộ nhớ quản lý hay cách thực thi bảo mật.

2.3 Tập ngôn ngữ JavaCard

Do JavaCard được tích hợp vào các bộ nhớ nhỏ của thiết bị lưu vết nên nền tảng của ngôn ngữ JavaCard hỗ trợ sự chọn lựa kỹ lưỡng từ tập ngôn ngữ của Java.

Dưới đây là bảng thống kê các thuộc tính mà thư viện Java hỗ trợ [6].

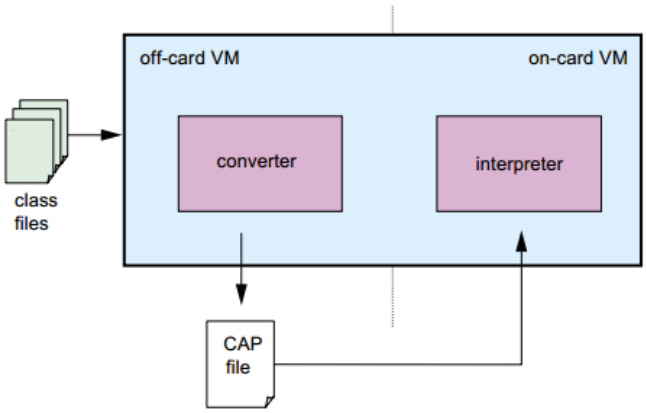
Bảng 2.1: Các thuộc tính mà thư viện hỗ trợ

Java hỗ trợ các tính năng	Java không hỗ trợ các tính năng
- Kiểu dữ liệu nhỏ: boolean, byte, short. - Mảng một chiều. - Gói java, lớp, giao diện, các ngoại lệ. - Các đặc tính hướng đối tượng trong Java: kế thừa, phương thức ảo, quá tải và tạo đối tượng động, chấp nhận phạm vi và quy tắc ràng buộc. - Các từ khóa kiểu nguyên (int) và 32-bit số nguyên kiểu dữ liệu hỗ trợ là tùy chọn.	- Kiểu dữ liệu lớn: long, double, float. - Kí tự và chuỗi. - Mảng đa chiều. - Class loading động. - Quản lý security. - Khởi tạo và hủy (bộ nhớ). - Luồng (Threads). - Object serialization (chuyển đổi trạng thái của một object). - Object cloning.

2.4 Máy ảo để chạy Java Card

Sự khác biệt chính giữa máy ảo JavaCard (JCVM) và máy ảo Java (JVM) đó là máy ảo JavaCard thực hiện thành hai phần: một phần chạy các ứng dụng trên thiết bị đầu cuối và phần còn lại chạy trên ứng dụng thẻ[6].

JCVM chỉ hỗ trợ một tập con giới hạn của ngôn ngữ lập trình Java, nhưng nó bảo tồn nhiều tính năng quen thuộc bao gồm các đối tượng, thừa kế, các gói, tạo đối tượng động, các phương pháp ảo, các giao diện và các ngoại lệ. JCVM giảm hỗ trợ cho một số yếu tố ngôn ngữ có thể sử dụng quá nhiều bộ nhớ làm hạn chế thẻ thông minh.



Hình 2.3 Máy ảo JavaCard[6]

Máy ảo trên thẻ (on – card) interpreter (hay còn gọi là trình thông dịch JavaCard) cung cấp hỗ trợ thời gian chạy mô hình ngôn ngữ Java và cho phép độc lập phần cứng.

Phần máy ảo Java trên các thiết bị đầu cuối (off – card), nó chứa công cụ JCC (javacard converter) hỗ trợ việc xác thực, đóng gói, tối ưu mã hóa thường được gọi là công cụ chuyển đổi thẻ Java.

Cap File: chứa một biểu diễn nhị phân thực thi các lớp trong một package Java. Một tệp CAP là một tệp JAR – là định dạng chứa tệp CAP, nó chứa một bộ thành phần được lưu trữ dưới dạng tệp cá nhân trong tệp JAR [6].

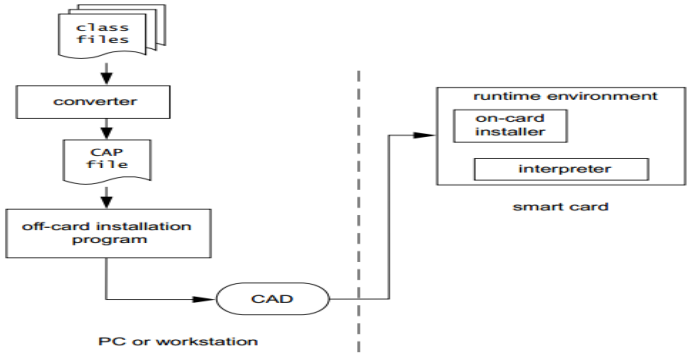
2.5 Cài đặt Java Card và chương trình cài đặt trên thiết bị (Off-Card)

Các công cụ sau đây yêu cầu cài đặt ứng dụng JavaCard:

- Một công cụ chuyển đổi để chuyển đổi một applet JavaCard sang một định dạng cần thiết để cài đặt.
- Các công cụ xác minh ngoài thẻ để kiểm tra tính toàn vẹn của các tệp được tạo ra bởi Converter.

Như đã đề cập trước đó, một applet JavaCard không cài đặt vào một thẻ thông minh, thay vì nó đã được cài đặt tệp CAP. Trình cài đặt thẻ ra tạo một tệp kịch bản chứa các APDU lệnh xác định phần đầu và kết thúc của tệp CAP, các thành phần và

dữ liệu thành phần. Tập tin kịch bản được sử dụng làm đầu vào cho tiện ích APDUTool. Tiện ích APDUTool đệ trình các lệnh APDU vào môi trường chạy JavaCard, hoặc đến một môi trường chạy mô phỏng như JCWDE.

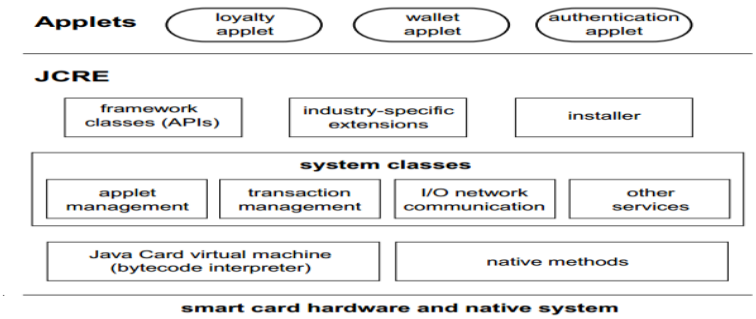


Hình 2.4 Trình cài đặt JavaCard và chương trình cài đặt ngoài thẻ.

2.6 Môi trường chạy JavaCard

Trình cài đặt và trình biên dịch không phải là thành phần hệ thống duy nhất chạy trên JavaCard, nó còn nhiều các thành phần khác. Môi trường chạy JavaCard (JCRE) bao gồm các thành phần hệ thống thẻ Java chạy bên trong một thẻ thông minh. JCRE chịu trách nhiệm về quản lý tài nguyên thẻ, truyền thông mạng, thực hiện applet và trên hệ thống thẻ và bảo mật applet. Do đó, nó thực chất là hệ điều hành của thẻ thông minh.

JCRE – nền tảng chạy ứng dụng JavaCard, nằm trên phần cứng của thẻ thông minh. JCRE bao gồm máy ảo JavaCard (trình thông dịch bytecode), các lớp khung ứng dụng JavaCard (API), các phần mở rộng chuyên ngành và các lớp hệ thống JCRE. Các JCRE độc đáo tách biệt Applet từ các công nghệ độc quyền của các nhà cung cấp thẻ thông minh và cung cấp hệ thống tiêu chuẩn và các giao diện API cho các applet. Kết quả là applet dễ viết và dễ di chuyển trên các kiến trúc thẻ thông minh khác nhau. Lớp dưới cùng của JCRE chứa máy ảo Java Card (JCVM) và method native (có tính địa phương). JCVM thực thi các bytecode, kiểm soát việc cấp phát bộ nhớ, quản lý các đối tượng và thực thi bảo mật thời gian chạy. Các method native (có tính địa phương) cung cấp hỗ trợ cho lớp JCVM và lớp hệ thống lớp tiếp theo. Họ có trách nhiệm xử lý các giao thức truyền thông cấp thấp, quản lý bộ nhớ, hỗ trợ mã hoá, v.v..[6].



Hình 2.5 Kiến trúc hệ thống trên thẻ.

.

JCRE được nạp vào thẻ Java tại nhà máy và vẫn duy trì ở đó cho đến khi thẻ bị phá hủy. Khi thẻ được đặt trong CAD (Card Accepting Device), nó được kích hoạt và bắt đầu sao chép dữ liệu từ chương trình từ EEPROM và ROM sang RAM nhanh hơn. Trong quá trình giao dịch, dữ liệu và các đối tượng phải được bảo toàn và được sao chép từ RAM vào EEPROM. EEPROM giữ dữ liệu khi không có điện, khi điện bị mất thẻ sẽ chuyển sang ngủ đông.

2.7 API Java Card

Vì các ứng dụng JavaCard chủ yếu liên quan đến nhiều công ty phát hành thẻ, tính tương thích phải được thiết kế ngay từ đầu. Từ quan điểm kỹ thuật, chìa khóa là một JavaCard API đây là một lớp phần mềm cho phép ứng dụng giao tiếp với thẻ thông minh và đầu đọc của nhiều nhà sản xuất.

JavaCard API bao gồm một tập lớp cho ứng dụng thẻ thông minh theo mô hình ISO 7816. API bao gồm ba package lõi và một package mở rộng. Ba package lõi đó là *java.lang*, *java.framework* và *java.security*. Package mở rộng đó là *java-cardx.crypto*. Có nhiều lớp nền tảng Java không hỗ trợ trong JavaCard API, ví dụ trong Java có một số class về GUI interface, network I/O và hệ thống file I/O màn hình là không được hỗ trợ trong JavaCard. Đó là lý do tại sao thẻ thông minh không có hiển thị nó sử dụng giao thức mạng và cấu trúc file hệ thống.

2.8 Package và quy ước đặt tên Applet

Hầu hết các ứng dụng quen thuộc được đặt tên và xác định bởi một tên chuỗi. Tuy nhiên, trong công nghệ JavaCard, mỗi applet được xác định và lựa chọn sử dụng một "định danh ứng dụng" (AID). Ngoài ra, mỗi gói Java được gán một AID. Điều này có nghĩa là một gói khi nạp vào một thẻ được liên kết với các gói khác đã được đặt trên thẻ thông qua AID của chúng. Quy ước đặt tên này phù hợp với đặc tả thẻ thông minh được định nghĩa trong ISO 7816.

Một AID là một mảng các byte có thể được hiểu là hai mảng riêng biệt, như trong bảng 2.3 phần thứ nhất là giá trị 5 byte được gọi là RID (mã nguồn). Phần thứ hai là một giá trị có độ dài biến được gọi là PIX (thuộc tính mở rộng định danh). PIX có thể từ 0 đến 11 byte chiều dài. Do đó một AID có thể từ 5 đến 16 byte trong tổng chiều dài. Định dạng của nó được mô tả trong:

Bảng 2.3 Cấu trúc ADI

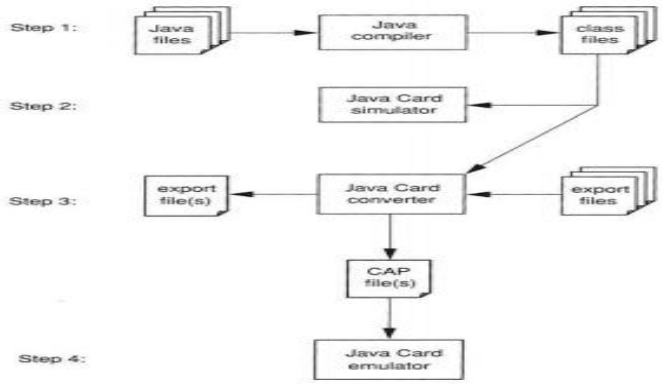
Định danh ứng dụng (AID)	
Nhà cung cấp ứng dụng đăng ký quốc gia (RID)	Mã nhận dạng ứng dụng độc quyền - mở rộng (PIX)
5 bytes	0 to 11 bytes

2.9 Java Card Applet

JavaCard applet không nên nhầm lẫn với các applet Java chỉ vì chúng đều là các applet được đặt tên. Một JavaCard Applet là một chương trình Java tuân thủ một tập hợp các quy ước cho phép nó chạy trong môi trường chạy Java Card. Một JavaCard Applet không được dự định chạy trong môi trường trình duyệt. Lý do tên applet đã được chọn cho các ứng dụng Java Card được nạp vào môi trường chạy Java Card sau khi thẻ đã được sản xuất. Tức là, không giống như các ứng dụng trong nhiều hệ thống nhúng, các applet không cần phải ghi vào ROM trong quá trình sản xuất. Thay vào đó, họ có thể tự động tải xuống thẻ sau đó.

2.9.1 Tiến trình phát triển Applet

Sự phát triển của một applet JavaCard bắt đầu cũng giống như với bất kỳ chương trình Java nào khác. Một nhà phát triển viết một hoặc nhiều lớp Java và biên dịch mã nguồn với một trình biên dịch Java, tạo ra một hoặc nhiều lớp. Hình 2.6 trình bày quá trình phát triển applet.



Hình 2.6 Tiến trình phát triển Applet[6]

2.9.2 Cài đặt applet

Việc cài đặt aplet được thực hiện tại nhà máy hoặc tại văn phòng của người phát hành và cũng có thực hiện sau khi phát hành, thông qua quá trình cài đặt an toàn (nếu nhà sản xuất thẻ xác định). Quá trình này bao gồm việc tải xuống một applet được ký kỹ thuật số, mà JCRE xác minh là hợp pháp, trước khi cài đặt applet. Các aplelet được cài đặt thông qua các tải không thể chứa các cuộc gọi phương thức tự nhiên do chúng không được tin cậy.

Để tải một applet, trình cài đặt thẻ sẽ thực hiện: thẻ sẽ lấy tệp CAP và chuyển đổi nó thành một chuỗi các lệnh APDU, mang nội dung tệp CAP. Bằng cách trao đổi các lệnh APDU với chương trình cài đặt không thẻ, trình cài đặt trên thẻ sẽ viết nội dung tệp CAP vào bộ nhớ liên tục của thẻ và liên kết các lớp trong tệp CAP với các class khác nằm trên thẻ. Trình cài đặt cũng tạo và khởi tạo bất kỳ dữ liệu nào được sử

dụng nội bộ bởi JCRE để hỗ trợ applet này. Nếu applet yêu cầu một vài gói để chạy, mỗi tệp CAP được nạp vào thẻ.

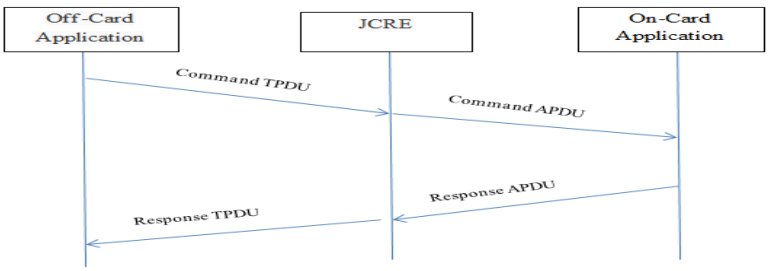
Bước cuối cùng trong quá trình cài đặt applet, trình cài đặt tạo ra một ví dụ applet và đăng ký với JCRE. Để làm như vậy, trình cài đặt sẽ gọi phương thức cài đặt:

public static void install (byte[] bArray, short offset, byte length)

Phương pháp cài đặt là một phương pháp của applet, tương tự như phương pháp chính trong một ứng dụng Java. Một applet phải thực hiện phương pháp cài đặt. Trong phương pháp cài đặt, nó gọi constructor của applet để tạo và khởi tạo một ví dụ applet. Các tham số cài đặt được gửi tới thẻ cùng với tệp CAP. Sau khi applet được khởi tạo và đăng ký với JCRE, nó có thể được chọn và chạy[6].

2.10 Phương thức truyền nhận, trao đổi dữ liệu

Ứng dụng Java Card thực hiện trao đổi dữ liệu với ứng dụng trên thiết bị đầu cuối thông qua đơn vị dữ liệu giao thức truyền tải(Transmission protocol data unit - TPDU) và đơn vị dữ liệu giao thức ứng dụng (Application protocol data unit - APDU). Luận văn tập trung vào xem xét cấu trúc của APDU và phương thức trao đổi TPDU, APDU trong hai giao thức giao tiếp T=0 và T=1 của JavaCard. Quy trình trao đổi TPDU, APDU giữa hai ứng dụng trên thẻ và trên thiết bị đầu cuối được thể hiện trong hình 2.7.



Hình 2.7 Trao đổi thông tin giữa ứng dụng trên thẻ và ứng dụng trên thiết bị đầu cuối

APDU

- *Cặp câu lệnh - phản hồi*

Một đơn vị dữ liệu giao thức ứng dụng là một câu lệnh APDU hoặc một phản hồi APDU. Một tiến trình trong giao thức ứng dụng bao gồm việc gửi lệnh APDU, xử lý nội dung nhận được và trả về APDU phản hồi. Hai APDU đó tạo thành cặp câu lệnh - phản hồi[6].

Bảng 2.4 Cấu trúc câu lệnh APDU

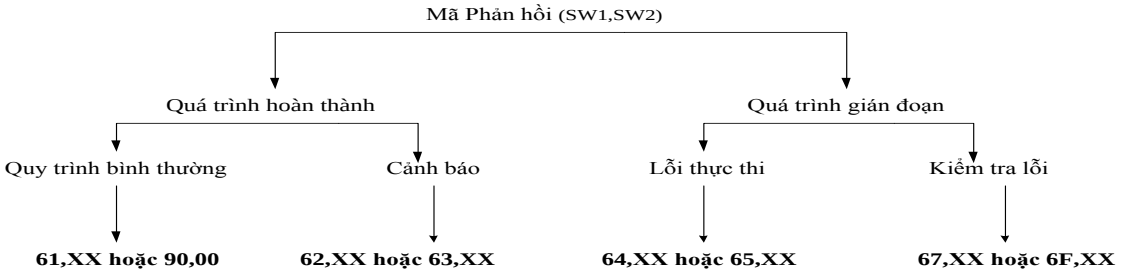
Command APDU						
Header (required)				**Body (optional)**		
CLA	INS	P1	P2	L_c	Data Field	L_e

Bảng 2.5 Cấu trúc APDU phản hồi

Response APDU		
Body (optional)	**Trailer (required)**	
Data Field	SW1	SW2

Câu lệnh APDU bắt buộc phải chứa phần mở đầu gồm 4 byte: CLA, INS, P1, P2 và có thể có thêm phần thân với độ dài tùy biến. APDU phản hồi bắt buộc phải chứa phần mã trạng thái (Status Word - SW) gồm 2 byte: SW1, SW2 và có thể có thêm phần thân với độ dài tùy biến.

Các giá trị của từ trạng thái được định nghĩa trong tiêu chuẩn ISO 7816-4:



Hình 2.8 Mã trạng thái phản hồi

CHƯƠNG 3 MẬT MÃ TRÊN ĐƯỜNG CONG ELLIPTIC

Vấn đề về đảm bảo An toàn thông tin đang được đặt lên hàng đầu trong bài toán giao dịch không an toàn trên Internet. Chúng ta cần có các giải pháp đảm bảo an toàn thông tin điều đó được xây dựng dựa trên lý thuyết mật mã, an toàn bảo mật thông tin. Các nhà khoa học đã phát minh ra những hệ mật mã như RSA, Elgamal, SHA1, SHA2, SHA3…nhằm che dấu thông tin cũng như là làm rõ chúng để tránh sự nhòm ngó của những kẻ cố tình phá hoại. Mặc dù rất an toàn nhưng có độ dài khoá lớn nên trong một số lĩnh vực không thể ứng dụng được. Chính vì vậy hệ mật trên đường cong elliptic ra đời, hệ mật này được đánh giá là hệ mật có độ bảo mật an toàn cao và hiệu quả hơn nhiều so với hệ mật công khai khác. Chương 3 trình bày những kiến thức về mật mã trên đường cong Elliptic.

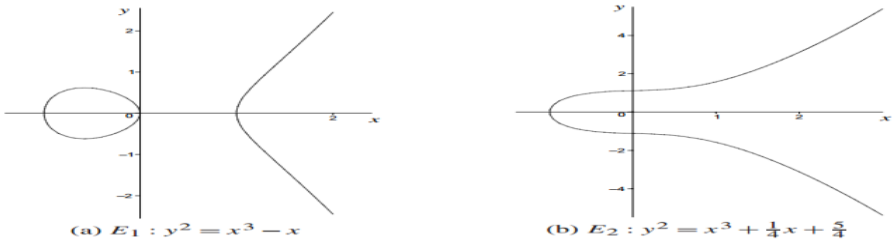
3.1 Cơ sở lý thuyết

a) Khái niệm đường cong Elliptic theo công thức Weierstrass

Đường cong elliptic E trên trường K là tập hợp các điểm (x,y) ∈ KxK thỏa mãn phương trình:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \ (a_i \in K \text{ và } 4a_4^3 + 27a_6^2 \neq 0)[4]$$

Một số ví dụ về đường cong Elliptic:



Hình 3.1 Một số ví dụ về đường cong Elliptic.

b) Đường cong Elliptic trên trường số thực R

Trên trường số thực R, đường cong Elliptic xác định bởi tập hợp các điểm (x,y) € R^2 thỏa mãn[4]:

$$y^2 = x^3 + ax + b, \text{ thỏa mãn điều kiện } 4a^3 + 27b^2 \neq 0$$

Các đa thức dạng $x^3 + ax + b$ có rất nhiều, do đó chúng ta phải chọn thêm một điểm *O* (điểm vô cực.

c) Đường cong elliptic trên trường F_p (p là số nguyên tố)

Cho p là số nguyên tố, (p>3). Một đường cong Eliptic trên trường F_p được định nghĩa bởi dạng:

$$y^2 = x^3 + ax + b, (1) \text{ Trong đó a, b} \in F_p \text{ và } 4a^3 + 27b^2 \not\equiv 0 \ (mod\ p).$$

Tập E(F_p) bao gồm tất cả các cặp điểm (x, y), với x, y ∈ F_p thỏa mãn phương trình (1) cùng với một điểm O – gọi là điểm tại vô cực.

S = {(x,y): $y^2 = x^3 + ax + b$, x, y ∈ F_p}∪ {O}[4].

Số lượng điểm của E(F_p) là #E(F_p) thỏa mãn định lý Hasse:

$$p + 1 - 2\sqrt{p} \leq \#E(Fp) \leq p + 1 + 2\sqrt{p}$$ [4].

d) Đường cong Elliptic trên trường hữu hạn F_{2^m}

Một đường cong Eliptic E trên trường F_{2^m} được xác định bởi phương trình có dạng:

$$y^2 + xy = x^3 + ax^2 + b, (2)$$

Trong đó a, b ∈ F_{2^m}, và b≠0. Tập E(F_{2^m}) bao gồm tất cả các điểm (x,y), x, y ∈ F_{2^m} thỏa mãn phương trình (2) cũng với điểm O là điểm tại vô cực[4].

S = {(x,y): $y^2 + xy = x^3 + ax^2 + b$, x, y ∈ F_{2^m}}∪ {O}.

Số lượng điểm của E(F_{2^m}) là #E(F_{2^m}) thỏa mãn định lý Hasse:

$$p + 1 - 2\sqrt{p} \leq \#E(F_P) \leq p + 1 + 2\sqrt{p}.$$

Trong đó $p = 2^m$. Ngoài ra #E(F_{2^m}) là số chẵn.

a. Các phép toán trên đường cong Elliptic

Phép cộng:

Giả sử P(x_1,y_1) và Q(x_2,y_2) là hai điểm của E. Nếu x_1=x_2 và y_1=-y_2 thì ta định nghĩa P + Q =O. Ngược lại thì P + Q = (x_3,y_3) ∈ E trong đó : $x_3=\lambda^2 - x_1 - x_2$, $y_3 = \lambda(x_1 - x_3) - y_1$. Với: P + O = O + P = P, ∀ P ∈ E(F_p)

Nếu P = (x, y) ∈ E(F_p), thì (x,y) + (x,-y) = O. Điểm (x, -y) được ký hiệu là -P và –P ∈ E(F_p)

Cho hai điểm P =(x_1, y_1) và Q = (x_2, y_2) ∈ E(F_p), nếu $P \neq \pm Q$ thì P+Q=(x_3,y_3) được xác định như sau:

Đặt $\lambda = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)$

$$x_3 = \lambda^2 - x_1 - x_2$$

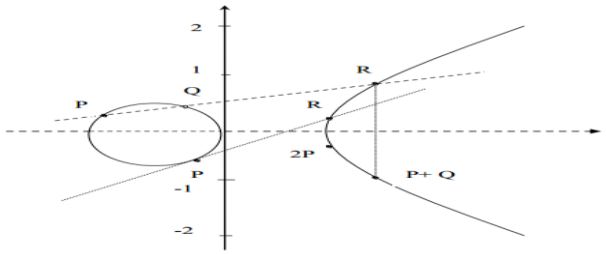
$$y_3 = \lambda(x_1 - x_3) - y_1.$$

Cho P =(x_1, y_1) ∈ E(F_p), P≠ -P. Khi đó 2P = (x_3,y_3) và được xác định như sau:

$x_3 = \lambda^2 - x_1 - x_2$

$y_3 = \lambda(x_1 - x_3) - y_1$

P = Q và y_1 = 0 thì P + Q = *O[4]*.



Hình 3.2 Phép cộng trên đường cong elliptic

- Phép nhân

Nếu cộng hai điểm P, Q ∈ E(R) với P = Q thì đường thẳng L sẽ là tiếp tuyến của đường cong elliptic tại điểm P. Trường hợp này điểm –R sẽ là giao điểm còn lại của L với E. Lúc đó R = 2P. Phép nhân Kp nhận được bằng cách thực hiện lặp k lần phép cộng[4].

3.2 Những chú ý để lựa chọn đường cong Elliptic phù hợp

Việc chọn một đường cong elliptic thế nào ảnh hưởng đến tốc độ, tính hiệu quả, độ dài khóa và tính an toàn của hệ mật mã trên đường cong này. Dù *E, K* và điểm cơ sở $P \in E$ cố định và công khai nhưng việc chọn các tham số này phù hợp là bước quan trọng nhất.

3.2.1 Trường K

Trước hết chúng ta xem xét sự ảnh hưởng của trường *K* đến cấu trúc nhóm của *E*(*K*) và các hệ mật mã trên *E*(*K*). Một đường cong elliptic trên một trường hữu hạn tạo thành nhóm Abel được sử dụng trong mật mã học.

3.2.2 Dạng của đường cong elliptic

Trước hết, chúng ta cần xem các dạng đường cong elliptic. Có 2 lớp đường cong elliptic được dùng trong các hệ mã hóa.

- Supersingular Curve

- Nonsupersingular

3.2.3 Phương pháp lựa chọn

Có một vài phương pháp để lựa chọn các đường cong elliptic. Phương pháp tự nhiên nhất là chọn ngẫu nhiên. Chọn ngẫu nhiên một đường cong elliptic *E* trên trường *K* và một điểm cơ sở $P \in E$. *K* được chọn và cố định trước. Phương pháp chọn ngẫu nhiên Koblitz cho các đường cong elliptic trên trường F_q (với *q* lớn).

3.3 So sánh RSA và ECC

Các so sánh tập trung vào 3 đặc điểm:

✓ **Độ an toàn (Security)**. Hệ mật đó đã được sử dụng rộng rãi bao lâu và tính an toàn của nó đã được nghiên cứu thế nào?

✓ **Tính hiệu quả (Efficiency)**. Độ phức tạp tính toán khi thực hiện.

✓ **Không gian lưu trữ (Space requirements)**. Không gian cần thiết để lưu trữ khóa cũng như các tham số khác của hệ mặt đó.

Độ an toàn

Tính hiệu quả

Không gian lưu trữ

3.4 Mật mã trên đường cong elliptic

Hệ mật dựa trên đường cong Elliptic (ECDSA/ECC) là một giải thuật khoá công khai. Hiện nay, hệ mật RSA là giải thuật khoá công khai được sử dụng nhiều nhất, nhưng hệ mật dựa trên đường cong Elliptic (ECC) có thể thay thế cho RSA bởi mức an toàn và tốc độ xử lý cao hơn.

3.5 Chữ ký số trên hệ mật đường cong Elliptic

3.5.1 Sơ đồ chữ ký ECDSA

[2]Để thiết lập sơ đồ chữ ký ECDSA (Elliptic Curve Digital Signature Algorithm), cần xác định các tham số: lựa chọn đường cong E trên trường hữu hạn Fq với đặc số p sao cho phù hợp, điểm cơ sở G ∈ E(Fq).

a. Sinh khóa

b. Ký trên bản rõ m

c. Kiểm tra chữ ký

d. Xác thực

3.5.2 Sơ đồ chữ ký Nyberg – Rueppel

[2]Giả sử E là một đường cong Elliptic trên trường Zp (p>3 và nguyên tố) sao cho E chứa một nhóm con cyclic H trong đó bài toán logarith rời rạc là "khó".

Với $P=Z_p^* x Z^*_p$, $C = E x Z_p^* x Z_p^*$ ta định nghĩa K={(E,Q,a,R):R=aQ} với Q ∈ E . Các giá trị α và R là công khai, a là bí mật.

Với K=(E,Q,a,R) chọn số ngẫu nhiên $k \in Z_{|H|}$. Khi đó, với $x = (x_1,x_2) \in Z_p^* x Z_p^*$ ta định nghĩa $sig_k(x,k) = (c,d)$.

3.5.3 Sơ đồ chữ ký mù Harn trên ECC[2]

a. Sinh khóa:

b. Ký mù

3.5.4 Sơ đồ chữ ký mù bội Harn trên ECC[2]

a. Sinh khóa

b. *Ký mù trên m*:

3.6 Đánh giá các tấn công hệ mật trên đường cong Elliptic

3.6.1 Phương pháp Baby step - Giant step

Phương pháp Babystep – GiantStep là phương pháp tấn công đầu tiên lên hệ mật mã ECC, và thực hiện với thời gian là hàm mũ. Nó giải bài toán DLP trong trường nguyên tố Zp được mở rộng cho bài toán EDLP. Bài toán Tìm k sao cho kG = Q trên E(Fq) với #E(Fq) = N, giả sử k tồn tại thực sự.

- *Đánh giá*: đối với các nhóm điểm đường cong elliptic cấp N, phương pháp này cần khoảng $\sqrt{N}$ bước tính và $\sqrt{N}$ bộ nhớ.

3.6.2 Phương pháp Pohlig – Hellman

Định nghĩa: Giả sử $p-1=\prod_{i-1}^{k} p_i^{Ci}$, p_i là số nguyên tố đặc biệt. Giá trị a = $\log_\alpha \beta$ được xác định một cách duy nhất theo modulo p-1. Trước hết nhận xét rằng, nếu có thể tính a mod p_i^{Ci} với mỗi i, $1 \le i \le k$, thì:

$$p\text{-}1 \equiv 0 \pmod{q^c}$$

$$p\text{-}1 \neq 0 \pmod{q^{c+1}}$$

có thể tính a mod (p-1) theo định lý phần dư. Để thực hiện diều đó ta giả sử rằng q là số nguyên tố.

Thuật toán Pohlig - Hellman để tính log$\alpha\beta$ mod qc

- Phương pháp Pohig – Hellman nó thực hiện tốt tất cả các ước nguyên tố của N là nhỏ. Nếu ước nguyên tố lớn nhất xấp xỉ độ lớn của N thì phương pháp này rất khó áp dụng. Do đó các hệ mật dựa trên logrith rời rạc thường chọn bậc của nhóm có chứa một thừa số nguyên tố lớn.

3.6.3 Tấn công MOV

Tấn công MOV chỉ ra không phải loại đường cong elliptic nào cũng có thể được sử dụng, bằng cách đưa ra việc giải quyết bài toán logarit rời rạc trên đường cong elliptic trở thành bài toán logarit rời rạc trên một trường hữu hạn mở rộng với độ mở rộng tùy thuộc vào loại đường cong. Do đó, các đường cong siêu lạ rất được ưa dùng ở giai đoạn đầu lại trở nên rất yếu vì ở đó độ mở rộng chỉ cao nhất là 6. Nét đặc biệt trong tấn công MOV là việc sử dụng phép ghép cặp (pairings) trên các đường cong elliptic, với những kết quả khá mới mẻ trong lý thuyết số. Không chỉ phục vụ cho việc phá mã, việc sử dụng phép ghép cặp sau đó đã trở nên cực kỳ hữu hiệu trong việc xây dựng mã.

3.6.4 Phương pháp Index và Xedni

Thuật toán tính chỉ số ngược đầu tiên là nâng các điểm P_1 , P_2 ,..., P_n , sau đó chọn một đường cong Elliptic E (Q) chứa các điểm đã nâng và hy vọng rằng chúng phụ thuộc tuyến tính. Nghĩa là thỏa mãn quan hệ $\sum_{i=1}^{r} ni\ Pi \ \ =0$. Tuy nhiên, xác suất để chúng phụ thuộc tuyến tính là nhỏ.

Thuật toán Index và Xedni có thời gian chạy là hàm mũ và không hiệu quả trong thực tế. Do thuật toán Index và Xedmi vướng phải hai bài toán khó bởi vì:

- Bài toán tìm được phép nâng theo nghĩa tạo ra các điểm nâng phụ thuộc tuyến tính bài toán này khó do xác suất các điểm nâng phụ thuộc tuyến tính là rất nhỏ.
- Giải phương trình quan hệ tuyến tính cũng rất khó vì độ cao của các điểm nâng là rất lớn.

3.6.5 Các tấn công dựa trên giả thuyết Diffie – Hellman

Tấn công này chỉ ra rằng nếu p-1 có một ước d thỏa mãn $\approx \sqrt{p}$ thì khóa bí mật có thể được tính với O($\sqrt[4]{p}$) bộ nhớ. Nếu p+1 có một ước d thỏa mãn d$\approx \sqrt[3]{p}$ thì khóa bí mật có thể được tính là O(log p.$\sqrt[3]{p}$) phép toán sử dụng O($\sqrt[3]{p}$) bộ nhớ.

3.6.6 Các tấn công cài đặt

Kiểu tấn công cài đặt thứ nhất là dựa trên điểm không hợp lệ của đường cong Elliptic. Nó được áp dụng trong một số giao thức cụ thể như mã hóa tích hợp đường cong Elliptic hoặc giao thức trao đổi khóa ECDH một pha. Nếu trong quá trình nhận và xử lý một điểm trên đường cong mà không thực hiện việc kiểm tra xem nó có thực sự nằm trên đường cong đã cho hay không thì lược đồ có thể bị tấn công.

Dạng tấn công thứ hai là kiểu tấn công phân tích năng lượng để khám phá khóa bí mật. Hiệu quả của các kiểu tấn công này phụ thuộc vào cách cài đặt cụ thể.

3.7 Chuẩn tham số cho hệ mật Elliptic

Trên thế giới hiện nay các chuẩn tham số cho hệ mật Elliptic được đưa ra trong các chuẩn:

- ISO 15496-5
- ANSI X9.62
- FIPS PUB 186-3
- Certicom SEC1 version 2.0

3.8 Sinh tham số cho hệ mật Elliptic

3.8.1 Tham số miền của đường cong Elliptic

Thuật toán sinh tham số miền của đường cong Elliptic[4]:

3.8.2 Sinh và kiểm tra cặp khóa đường cong Elliptic

Thuật toán : Sinh cặp khóa cho hệ mật Elliptic

+ Input: Bộ tham số miền(Fp, A,B,G,N,h, SEED)

+ Output: Output: (Q – điểm công khai, d – khóa bí mật)

Thuật toán: Kiểm tra tính hợp lệ của khóa công khai

+ Input: Tham số miền (F_p , A, B , G , N , h, $SEED$), khóa công khai Q

+ Output: "*Khóa công khai hợp lệ*" hoặc "*Khóa công khai không hợp lệ*"

3.8.3 Thuật toán kiểm tra điều kiện MOV

Thuật toán:

+ Input: Giá trị B là cận của MOV theo tiêu chuẩn EC5

+ Output: 0: Không thỏa mãn điều kiện MOV; 1: Thỏa mãn MOV.

3.8.4 Thuật toán sinh đường cong ngẫu nhiên

Thuật toán : Sinh đường cong ngẫu nhiên

+ Input: Số nguyên tố p

+ Output: Chuỗi SEED và A,B$\in$ F_p Xác định E trên F_p

CHƯƠNG 4 ỨNG DỤNG CHỮ KÝ SỐ TRÊN ĐƯỜNG CONG ELLIPTIC NHẰM ĐẢM BẢO ATTT TRONG ĐĂNG KÝ THẺ TRỰC TUYẾN

4.1 Bài toán

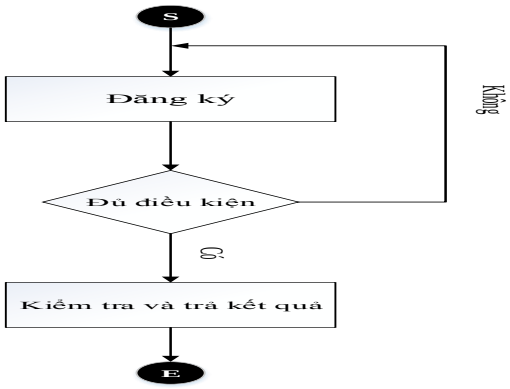
Việc phát triển thanh toán, mua hàng online bằng thẻ Ngân hàng là xu hướng phát triển tất yếu tại thị trường việt nam hiện nay. Phát triển thanh toán bằng thẻ online cũng là sự đầu tư đúng đắn theo chủ trương hạn chế giao dịch bằng tiền mặt của Ngân hàng Nhà nước. Thêm vào đó, sự đầu tư và cam kết về chất lượng của nhiều ngân hàng sẽ là điểm dựa đáng tin cậy cho các đối tượng khách hàng sử dụng thanh toán online. Tối ưu hóa mọi giao dịch với chiếc thẻ ngân hàng và cú nhấp chuột để tận hưởng cuộc sống một cách đơn giản, tiện lợi và thoải mái nhất.

Để có được những lợi ích mang lại của thẻ thông minh thì người dùng phải đăng ký để sỡ hữu được thẻ thông minh cho riêng mình. Người dùng chuẩn bị giấy tờ như chứng minh thư, ảnh, thông tin chứng minh thu nhập, mang giấy tờ đến tại chi nhánh ngân hàng muốn đăng ký mở thẻ. Sau khi ngân hàng xác nhận thông tin hợp lệ sẽ tiến hành cấp thẻ sau năm đến bẩy ngày làm việc. Thủ tục đăng ký thẻ rườm rà, mất thời gian. Người dùng cũng phải mất công chạy đi chạy lại ngân hàng để tiến hành làm thủ tục đăng kí, thời gian đăng ký cũng hạn chế trong giờ hành chính gây bất tiện cho người đăng ký thẻ mới. Ngoài ra thời gian chờ đợi thẻ cũng mất 7 ngày và phải lên đúng chi nhánh nơi mình đã đăng ký để nhận thẻ. Hiện nay có một số ngân hàng có hình thức đăng ký thẻ trực tuyến tuy nhiên việc bảo mật thông tin cho khách hàng đang còn lỏng lẻo dẫn đến mất an toàn thông tin. Vấn đề đặt ra phải đảm bảo An toàn thông tin trong giao dịch trực tuyến và đăng ký thẻ. Cần đưa các hệ mật an toàn vào quá trình mã hóa, giải mã, cũng như chứng thực chứng từ liên quan trong quá trình đăng ký cũng như giao dịch trên mạng Internet không an toàn. Hệ mật dựa trên đường cong Elliptic được đánh giá là hệ mật có độ bảo mật an toàn cao và hiệu quả hơn nhiều so với hệ mật công khai khác. Do đó trong phạm vi luận văn này đề xuất áp dụng hệ mật trên đường cong Elliptic trong quá trình đăng ký thẻ tín dụng trực tuyến.

4.2 Giải pháp kết hợp chữ ký ECDSA trong đăng ký thẻ trực tuyến

4.2.1 Quy trình đăng ký thẻ trực tuyến

Trong hệ thống đăng ký thẻ trực tuyến, người dùng phải thực hiện các bước bao gồm: đăng ký thẻ; ngân hàng thực hiện các bước: kiểm tra và trả kết quả. Quy trình được mô tả như hình:



Hình 4.1 Quy trình đăng ký thẻ trực tuyến.

Đăng ký: Trong bước này người dùng tiến hành đăng ký các thông tin trên hệ thống đồng thời cung cấp các giấy tờ liên quan.

Kiểm tra và trả kết quả: Ngân hàng xác minh tính hợp lệ của tờ khai và các giấy tờ cung cấp sau đó sẽ trả kết quả.

4.2.2 Chữ ký ECDSA dùng trong đăng ký thẻ trực tuyến.

Để bảo mật thông tin cho khách hàng khi đăng ký thẻ trực tuyến chúng ta sẽ ứng dụng chữ ký ECDSA vào quá trình đăng ký thẻ trực tuyến để đảm bảo rằng chính người ký là người tạo ra nó, không thể làm giả chữ ký nếu như không biết thông tin bí mật để tạo chữ ký, một khi đã ký thì người ký không thể phủ nhận chữ ký đó.

Lý do chọn chữ ký ECDSA bởi vì ưu điểm độ an toàn của nó, độ an toàn của chữ ký ECDSA dựa trên bài toán logarit rời rạc trên đường cong elliptic. Cho đến nay độ an toàn của các hệ mã hoá đường cong elliptic đã được chỉ ra là rất an toàn và hiệu quả. Thuật toán giải bài toán logarit rời rạc đường cong elliptic tốt nhất hiện nay là thuật toán Pollard’s Rho, phiên bản thiết kế theo hướng tính toán song song.

Sơ đồ khối chữ ký ECDSA:

Quy trình ký và kiểm tra tính toàn vẹn của đăng ký thẻ trực tuyến:

4.2.3 Thiết kế chương trình

Bước 1: Ngân hàng sẽ gửi một bản đăng ký cho người dùng bao gồm các thông tin:

Khách hàng sẽ thực thiện kê khai thông tin và cung cấp giấy tờ đi kèm.

Bước 2: Tiến hành mã hóa rồi ký điện tử văn bản và gửi đến ngân hàng

Bước 3: Ngân hàng sẽ tiến hành giải mã và xác thực chữ ký. Kiểm tra thông tin trên mẫu khai. Nếu thông tin hợp lệ sẽ tiến hành các thủ tục đăng ký theo quy trình nghiệp vụ bên ngân hàng. Và gửi lại kết quả trong thời gian nhanh nhất qua bưu điện hoặc chuyển phát nhanh.

KẾT LUẬN

Các kết quả đã đạt được

Thẻ thông minh đã và đang được phát triển mạnh mẽ không chỉ trên thế giới mà tại việt nam thẻ thông minh cũng đang ngày càng sôi động, hứa hẹn tạo một bước ngoặt mới cho thị trường thẻ với những ứng dụng và tiện ích vô cùng độc đáo. Ngoài các cơ hội là những thách thức không hề nhỏ, đó chính là vấn đề bảo mật thông tin ngày nay đang đặt lên hàng đầu.

Trong khóa luận này tôi đã trình bày được những kết quả sau:

+ Giới thiệu tổng quan về thẻ thông minh: khái quát lịch sử phát triển của thẻ thông minh, nêu lên cấu tạo và phân loại thẻ. Phân tích chi tiết về ưu nhược điểm của thẻ thông minh, ngoài ra thách thức trong việc phát triển thẻ thông minh.

+ Nghiên cứu về công nghệ Java Card:

- Giới thiệu về JavaCard, kiến trúc của nó, tập ngôn ngữ.
- Trình bày về máy ảo để chạy, môi trường chạy, api java card, quy ước đặt tên, ứng dụng applet
- Trình bày về các giao thức truyền nhận dữ liệu giữa thẻ và thiết bị đầu cuối

+ Mật mã đường cong Elliptic

- Trình bày cơ sở lý thuyết đường cong Elliptic
- Mật mã đường cong
- Chữ ký số trên hệ mật đường cong elliptic.
- Đánh giá tấn công trên hệ mật elliptic
- Chuẩn tham số cho hệ mật
- Cách sinh tham số cho hệ mật

+ Ứng dụng chữ ký số trên đường cong Elliptic nhằm đảm bảo ATTT trong đăng ký thẻ trực tuyến.

- Sử dụng chữ ký điện tử trên hệ mật đường cong Elliptic
- Demo được chương trình.

HƯỚNG NGHIÊN CỨU TIẾP THEO

- Tìm hiểu cải tiến công nghệ JavaCard ứng dụng vào thẻ thông minh để nâng cao tính bảo mật cho thẻ thông minh.