Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

**XÂY DỤNG ỨNG DỤNG GIAO TIẾP USB TRÊN WINDOWS VÀ ỨNG DỤNG TRONG DỊCH VỤ CHỮ KÝ SỐ**

|  |  |
| --- | --- |
| Sinh viên thực hiện | Hoàng Viết Huy |
| Người hướng dẫn | ThS. Phạm Ngọc Hưng |

­­

Hà Nội, 04/2019

**PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

**1. Thông tin về sinh viên**

Họ và tên sinh viên: Hoàng Viết Huy

Điện thoại liên lạc: 0976986858 Email: hvhuybk@gmail.com.

Lớp: An toàn thông tin K59 Hệ đào tạo: Đại học chính quy

Thời gian làm ĐATN: Từ ngày 23/01/2019 đến 19/05/2019

**2. Mục đích nội dung của ĐATN**

* Tìm hiểu việc firmware cho dòng thiết bị tương ứng
* Xây dựng một ứng dụng Windows cho phép ký file bằng chữ ký từ thiết bị.

**3. Các nhiệm vụ cụ thể của ĐATN**

* Tìm hiểu việc lập trình firmware cho dòng thiết bị tương ứng.
* Tìm hiểu giao thức giao tiếp USB giữa vi điều khiển và PC Windows.
* Tìm hiểu lập trình Winform giao diện trên Windows
* Tìm hiểu phương pháp ký tài liệu bằng chữ ký điện tử.
* Tổng kết và đánh giá.

**4. Lời cam đoan của sinh viên:**

Tôi – Hoàng Viết Huy - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS Phạm Ngọc Hưng*.

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 24 tháng 05 năm 2018*  Tác giả ĐATN  *Hoàng Viết Huy* |

**5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:**

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 24 tháng 05 năm 2018*  Giáo viên hướng dẫn  *ThS Phạm Ngọc Hưng* |

**LỜI CẢM ƠN**

Để có thể hoàn thành đồ án tốt nghiệp này, em xin gửi lời cảm ơn chân thành nhất tới tập thể các thầy giáo, cô giáo trường Đại học Bách Khoa Hà Nội nói chung, viện Công nghệ thông tin và truyền thông nói riêng, đã đào tạo cho em những kiến thức, những kinh nghiệm quý báu trong suốt thời gian học tập và rèn luyện.

Em xin gửi lời cảm ơn tới thầy giáo, ThS. Phạm Ngọc Hưng - Giảng viên bộ môn Kỹ thuật máy tính, viện Công nghệ thông tin và truyền thông, trường Đại học Bách Khoa Hà Nội đã tận tình hướng dẫn em trong quá trình làm đồ án tốt nghiệp.

Tiếp theo, em xin gửi lời cảm ơn chân thành tới toàn thể các anh chị trong công ty An ninh mạng Viettel đã giúp đỡ rất nhiệt tình và tạo mọi điều kiện tốt nhất trong suốt quá trình em làm và hoàn thành đồ án tốt nghiệp tại công ty.

Cuối cùng là lời cảm ơn chân thành nhất tới những người thân thân trong gia đình, bạn bè đã luôn ở bên động viên, góp ý và tạo mọi điều kiện tốt nhất để tôi có thể hoàn thành đồ án tốt nghiệp này.

Tuy nhiên, do thời gian và kiến thức có hạn nên đồ án này chắc chắn không tránh khỏi những thiếu sót, em rất mong được sự đóng góp ý kiến của các thầy, các cô và toàn thể các bạn. Em xin chân thành cảm ơn.

Hà Nội ngày 23 tháng 5 năm 2012

Sinh viên: Hoàng Viết Huy

# TÓM TẮT

Hiện nay, công nghệ thẻ thông minh có thể tạo ra một môi trường an toàn để lưu trữ khóa bí mật và thực thi các thao tác mật mã. Thiết bị USB Token dựa trên công nghệ thẻ thông minh tạo ra môi trường lưu trữ và tính toán cách ly an toàn và không thể nhân bản được nên không thể bị sao chép hoặc làm giả.

Token được sử dụng để kê khai nộp thuế trực tuyến, kê khai hải quan điện tử, giao dịch ngân hàng điện tử, giao dịch chứng khoán điện tử, Cổng thông tin một cửa quốc gia, cơ quan hành chính… mà không phải in các tờ kê khai, đóng dấu đỏ của công ty và gần đây token còn được sử dụng để giao dịch trong lĩnh vực kê khai bảo hiểm xã hội điện tử.

Từ nhu cầu trên, và mong muốn mang lại một sản phẩm có ý nghĩa và có tính ứng dụng cao đã dẫn em đến ý tưởng xây dựng thiết bị USB Token có các chức năng an toàn, mà không phụ thuộc vào các thiết bị có sẵn trên thị trường, giúp giảm giá thành sản phẩm.

Với những lý do trên, em quyết định chọn đồ án tốt nghiệp của mình là “**Xây dựng thiết bị USB Token**”.

Mục tiêu chính của đồ án là nghiên cứu công nghệ nhúng, lập trình nhúng, xây dựng kịch bản giao tiếp thiết bị với Windows. Ứng dụng công nghệ này có thể dùng cho các việc sử dụng khóa riêng cho các yêu cầu như bảo đảm tính bí mật và tính toàn vẹn của dữ liệu. Ứng dụng USBTokenManager dùng để đảm bảo tính toàn vẹn dữ liệu bằng cách sử dụng khóa bí mật để ký chữ ký điện tử lên các file văn bản, để đảm bảo chữ ký này là duy nhất và không thể giả mạo.

Nội dung chính đồ án bao gồm :

**PHẦN I: NỀN TẢNG VÀ CÔNG NGHỆ PHÁT TRIỂN HỆ THỐNG**

Chương này tập trung đề cập đến thực trạng việc sử dụng Token CA và sự cần thiết về thiết bị Token CA. Các giải pháp đưa ra để giải quyết thực trạng này.

Chương này thực hiện khảo sát các công nghệ nhúng dùng trong thiết bị Token CA hiện có dùng để xác thực các thông tin cá nhân của người dùng như: thẻ RFID, thẻ từ, thẻ thông minh, thiết bị USB... Chương này cũng đề xuất giải pháp sẽ sử dụng lưu khóa bí mật: thiết bị USB. Cuối cùng là khảo sát các linh kiện cần thiết có thể sử dụng để xây dựng thiết bị USB.

**Chương II: Thiết kế và xây dựng phần cứng**

* Chương này tập trung vào đưa ra khảo sát các thiết bị phần cứng hiện tại và đề xuất công nghệ nhúng sử dụng:
* Phần đầu chương giới thiệu, khảo sát các công nghệ nhúng dùng trong thiết bị Token CA hiện có như: thẻ RFID, thẻ từ, thẻ thông minh, thiết bị USB.
* Tiếp theo, đề xuất giải pháp sẽ sử dụng lưu khóa bí mật: thiết bị USB
* Khảo sát các linh kiện cần thiết có thể sử dụng để xây dựng thiết bị USB.
* Cuối cùng là thiết kế mạch cho thiết bị.

**Chương II: Xây dựng giải pháp và thiết kế phần sụn (Firmware)**

* Phần đầu của chương này sẽ tập trung khảo sát giải pháp giao tiếp giữa thiết bị và ứng dụng Windows, sau đó đề xuất giải pháp giao tiếp sử dụng WinUSB.
* Phần tiếp theo của chương sẽ xây dựng kịch bản giao tiếp giữa firmware và ứng dụng Windows.

**Chương III:** **Xây dựng giải pháp và kịch bản giao tiếp giữa firmware và ứng dụng trên Windows**

Phần đầu của chương này sẽ tập trung khảo sát giải pháp giao tiếp giữa thiết bị và ứng dụng Windows, sau đó đề xuất giải pháp giao tiếp sử dụng WinUSB.

Phần tiếp theo của chương sẽ xây dựng kịch bản giao tiếp giữa firmware và ứng dụng Windows.

**Chương IV: Xây dựng giao diện và demo một ứng dụng chữ ký điện tử.**

* Chương này tập trung vào việc đánh giá kết quả đạt được, đưa ra phương hướng phát triển cho hệ thống sau khi kết thúc đồ án và phần kết luận.

**MỤC LỤC**

[**PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP 2**](#_Toc515323278)

[**LỜI CẢM ƠN 3**](#_Toc515323279)

[**TÓM TẮT NỘI DUNG ĐỒ ÁN 4**](#_Toc515323280)

[**MỤC LỤC 5**](#_Toc515323281)

[**DANH MỤC CÁC BẢNG 7**](#_Toc515323282)

[**DANH MỤC HÌNH VẼ 8**](#_Toc515323283)

[**DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ 9**](#_Toc515323284)

[**PHẦN I: NỀN TẢNG VÀ CÔNG NGHỆ PHÁT TRIỂN HỆ THỐNG 10**](#_Toc515323285)

[1.1. Tổng quan 10](#_Toc515323286)

[1.1.1. Các vấn đề, khó khăn hiện tại 10](#_Toc515323287)

[1.1.2. Mục tiêu cần đạt được 11](#_Toc515323288)

[1.1.3. Lựa chọn và định hướng thiết kế 12](#_Toc515323289)

[1.2. Yêu cầu về thiết bị 13](#_Toc515323290)

[1.3. Công nghệ sử dụng 13](#_Toc515323291)

[1.3.1. Framework ASP.NET MVC 13](#_Toc515323292)

[1.3.2. Hệ quản trị CSDL SQL Server 15](#_Toc515323293)

[**PHẦN II: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 16**](#_Toc515323294)

[2.1. Mô hình vận hành của hệ thống mới 16](#_Toc515323295)

[2.2. Phân rã chức năng 17](#_Toc515323296)

[2.3. Usecase tổng quan 18](#_Toc515323297)

[2.3.1. Mô hình Use case tổng quan hệ thống 18](#_Toc515323298)

[2.4. Phân tích yêu cầu chức năng của hệ thống 20](#_Toc515323299)

[2.4.1. Chức năng đăng nhập (login): 20](#_Toc515323300)

[2.4.2. Chức năng tạo mới người dùng 21](#_Toc515323301)

[2.4.3. Chức năng quản lý đơn nhập kho (PO) 23](#_Toc515323302)

[2.4.4. Chức năng quản lý đơn xuất kho (PL) 25](#_Toc515323303)

[2.4.5. Chức năng nhập kho (Receipt) 26](#_Toc515323304)

[2.4.6. Chức năng xuất kho (Issue) 27](#_Toc515323305)

[2.4.7. Chức năng xuất báo cáo 28](#_Toc515323306)

[2.4.8. Chức năng quản lý vật tư 29](#_Toc515323307)

[2.5. Thiết kế hệ thống 33](#_Toc515323308)

[2.5.1. Sơ đồ thực thể liên kết 33](#_Toc515323309)

[2.5.2. Thiết kế CSDL 34](#_Toc515323310)

[2.5.3. Biểu đồ lớp 39](#_Toc515323311)

[2.5.4. Triển khai MVC 39](#_Toc515323312)

[2.5.5. Thiết kế giao diện 41](#_Toc515323314)

[**PHẦN III: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG 44**](#_Toc515323315)

[3.1. Thông tin sản phẩm 44](#_Toc515323316)

[3.2. Triển khai cơ sở dữ liệu 45](#_Toc515323317)

[3.3. Giao diện chức năng 45](#_Toc515323319)

[3.3.1. Trang chủ 45](#_Toc515323320)

[3.3.2. Đăng nhập 46](#_Toc515323321)

[3.3.3. Quản lý người dùng 47](#_Toc515323322)

[3.3.4. Quản lý quyền hạn 48](#_Toc515323323)

[3.3.5. Quản lý đơn đặt hàng 49](#_Toc515323324)

[3.3.6. Quản lý công ty 51](#_Toc515323326)

[3.3.7. Quản lý chi nhánh 52](#_Toc515323327)

[3.3.8. Quản lý kiện hàng 53](#_Toc515323328)

[3.3.9. Quản lý vật tư 54](#_Toc515323329)

[**PHẦN IV: KẾT LUẬN 55**](#_Toc515323330)

[**DANH MỤC TÀI LIỆU THAM KHẢO 56**](#_Toc515323331)

**DANH MỤC CÁC BẢNG**

[Bảng 1. Bảng mô tả UC chức năng đăng nhập 21](#_Toc515323451)

[Bảng 2. Bảng mô tả các bước sử dụng chức năng đăng nhập 21](#_Toc515323452)

[Bảng 3. Bảng mô tả UC tạo mới người dùng 22](#_Toc515323453)

[Bảng 4. Bảng mô tả các bước sử dụng chức năng đăng ký 22](#_Toc515323454)

[Bảng 5. Bảng mô tả UC thêm mới PO 24](#_Toc515323455)

[Bảng 6. Bảng mô tả các bước sử dụng chức năng thêm mới PO 24](#_Toc515323456)

[Bảng 7. Bảng mô tả UC thêm mới PL 26](#_Toc515323457)

[Bảng 8. Bảng mô tả các bước sử dụng chức năng thêm mới PL 26](#_Toc515323458)

[Bảng 9. Bảng mô tả UC chức năng nhập kho 27](#_Toc515323459)

[Bảng 10. Bảng mô tả các bước sử dụng chức năng nhập kho 27](#_Toc515323460)

[Bảng 11. Bảng mô tả UC chức năng xuất kho 28](#_Toc515323461)

[Bảng 12. Bảng mô tả các bước sử dụng chức năng xuất kho 29](#_Toc515323462)

[Bảng 13. Bảng mô tả UC chức năng xuất báo cáo 29](#_Toc515323463)

[Bảng 14. Bảng mô tả các bước sử dụng chức năng xuất kho 30](#_Toc515323464)

[Bảng 15. Bảng mô tả UC thêm mới vật tư 30](#_Toc515323465)

[Bảng 16. Bảng mô tả các bước sử dụng chức năng thêm mới PL 31](#_Toc515323466)

[Bảng 17. Bảng mô tả UC sửa vật tư 32](#_Toc515323467)

[Bảng 18. Bảng mô tả các bước sử dụng chức năng sửa vật tư 32](#_Toc515323468)

[Bảng 19. Bảng Công ty 35](#_Toc515323469)

[Bảng 20. Bảng Chi nhánh 35](#_Toc515323470)

[Bảng 21. Bảng Nhân viên 36](#_Toc515323471)

[Bảng 22. Bảng Quyền hạn 36](#_Toc515323472)

[Bảng 23. Bảng Chi tiết quyền hạn 36](#_Toc515323473)

[Bảng 24. Bảng Module quyền hạn 37](#_Toc515323474)

[Bảng 25. Bảng Vị trí lưu kho 37](#_Toc515323475)

[Bảng 26. Bảng Trạng thái vị trí 37](#_Toc515323476)

[Bảng 27. Bảng Kiện hàng 38](#_Toc515323477)

[Bảng 28. Bảng Trạng thái kiện hàng 38](#_Toc515323478)

[Bảng 29. Bảng Vật tư 38](#_Toc515323479)

[Bảng 30. Bảng Đơn nhập hàng 38](#_Toc515323480)

[Bảng 31. Bảng Chi tiết đơn nhập hàng 39](#_Toc515323481)

[Bảng 32. Bảng Đơn xuất vật tư 39](#_Toc515323482)

[Bảng 33. Bảng Chi tiết đơn xuất nhập tư 39](#_Toc515323483)

**DANH MỤC HÌNH VẼ**

[Hình 1. Minh họa nhập hàng hiện thời bằng phương pháp thủ công 11](#_Toc515323408)

[Hình 2. Mô hình MVC 14](file:///C:\Users\CUONGBK\Downloads\BaocaoDATN_NguyenVanCuong_20135181.docx#_Toc515323409)

[Hình 3. Mô hình hoạt động với website 17](#_Toc515323410)

[Hình 4. Sơ đồ phân rã chức năng 18](#_Toc515323411)

[Hình 5. Sơ đồ Use case tổng quan hệ thống 19](#_Toc515323412)

[Hình 6. Sơ đồ use case Cấu hình 20](file:///C:\Users\CUONGBK\Downloads\BaocaoDATN_NguyenVanCuong_20135181.docx#_Toc515323413)

[Hình 7. Sơ đồ Use case báo cáo 20](#_Toc515323414)

[Hình 8. Biểu đồ hoạt động chức năng đăng nhập 22](#_Toc515323415)

[Hình 9. Biểu đồ hoạt động chức năng tạo mới người dùng 23](#_Toc515323416)

[Hình 10. Biểu đồ tuần tự chức năng quản lý đơn nhập kho và quản lý nhập kho 24](#_Toc515323417)

[Hình 11. Biểu đồ hoạt động chức năng thêm mới PO 25](#_Toc515323418)

[Hình 12. Biểu đồ tuần tự chức năng quản lý đơn xuất kho và quản lý xuất kho 26](#_Toc515323419)

[Hình 13. Hình biểu đồ hoạt động chức năng thêm mới PL 27](#_Toc515323420)

[Hình 14. Hình biểu đồ hoạt động chức năng nhập kho 28](#_Toc515323421)

[Hình 15. Hình biểu đồ hoạt động chức năng nhập kho 29](#_Toc515323422)

[Hình 16. Biểu đồ chức năng xuất báo cáo 30](#_Toc515323423)

[Hình 17. Hình biểu đồ hoạt động chức năng thêm mới vật tư 32](#_Toc515323424)

[Hình 18. Hình biểu đồ hoạt động chức năng sửa vật tư 34](#_Toc515323425)

[Hình 19. Sơ đồ thực thể liên kết 35](#_Toc515323426)

[Hình 20. Các lớp Models 41](file:///C:\Users\CUONGBK\Downloads\BaocaoDATN_NguyenVanCuong_20135181.docx#_Toc515323427)

[Hình 21. Các lớp controllers 41](file:///C:\Users\CUONGBK\Downloads\BaocaoDATN_NguyenVanCuong_20135181.docx#_Toc515323428)

[Hình 22. Một số Views 41](file:///C:\Users\CUONGBK\Downloads\BaocaoDATN_NguyenVanCuong_20135181.docx#_Toc515323429)

[Hình 23. Giao diện Trang chủ 42](#_Toc515323430)

[Hình 24. Giao diện Danh sách đơn đặt hàng 42](#_Toc515323431)

[Hình 25. Giao diện Nhập kho 43](#_Toc515323432)

[Hình 26. Giao diện Quản lý kiện hàng 43](#_Toc515323433)

[Hình 27. Giao diện In barcode kiện hàng 44](#_Toc515323434)

[Hình 28. Giao diện Quản lý công ty 44](#_Toc515323435)

[Hình 29. Triển khai cơ sở dữ liệu 46](#_Toc515323436)

[Hình 30. Trang chủ 46](#_Toc515323437)

[Hình 31. Đăng nhập 47](#_Toc515323438)

[Hình 32. Danh sách người dùng 48](#_Toc515323439)

[Hình 33. Thêm mới người dùng 48](#_Toc515323440)

[Hình 34. Danh sách quyền hạn 49](#_Toc515323441)

[Hình 35. Quản lý quyền hạn chi tiết 50](#_Toc515323442)

[Hình 36. Danh sách đơn đặt hàng 50](#_Toc515323443)

[Hình 37. Tạo mới đơn đặt hàng 51](#_Toc515323444)

[Hình 38. Danh sách công ty 52](#_Toc515323445)

[Hình 39. Thêm mới công ty 53](#_Toc515323446)

[Hình 40. Danh sách chi nhánh 53](#_Toc515323447)

[Hình 41. Thêm mới chi nhánh 54](#_Toc515323448)

[Hình 42. Danh sách kiện hàng 54](#_Toc515323449)

[Hình 43. Danh sách vật tư 55](#_Toc515323450)

**DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ**

|  |  |  |
| --- | --- | --- |
| **Số thứ tự** | **Từ viết tắt** | **Ý nghĩa** |
| 1 | ĐATN | **Đ**ồ **Á**n **T**ốt **N**ghiệp |
| 2 | OS | **O**perating **S**ystem – Hệ điều hành |
| 3 | PO | **P**urchase **O**rder – Đơn đặt hàng |
| 4 | PL | **P**ick **L**ist – Đơn lấy hàng |
| 5 | WMS | **W**arehouse **M**anagement **S**ystem |
| 6 | CSDL | **C**ơ **S**ở **D**ữ **L**iệu |
| 7 | LTV | **L**ập **T**rình **V**iên |
| 8 | SQL | **S**tructured **Q**uery **L**anguage |
| 9 | UC | **U**se **C**ase |
| 10 | RDBMS | **R**elational **D**atabase **M**anagement **S**ystem |

**1**

**CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI**

Trong chương này sẽ giới thiệu tổng quan, mục tiêu cần đạt được, lựa chọn kỹ thuật để giải quyết mục tiêu đề ra.

## 1.1 Tổng quan

USB (Universal Serial Bus) là một chuẩn kết nối tuần tự đa dụng trong máy tính. USB sử dụng để làm một chuẩn kết nối các thiết bị ngoại vi với máy tính, chúng thường được thiết kế dưới dạng các đầu cắm cho các thiết bị tuân theo chuẩn cắm-và-chạy mà với tính năng cắm nóng thiết bị (nối và ngắt các thiết bị không cần phải khởi động lại hệ thống).

Giao tiếp USB được ứng dụng trong nhiều thiết bị với giao thức đã được chuẩn hóa. Tuy nhiên, với các nhu cầu riêng biệt, thì yêu cầu cần các giao thức không theo chuẩn là nhu cầu cần thiết. Từ đó, việc hiểu về cách thức giao tiếp USB là rất quan trọng để có thể tùy chỉnh theo từng nhu cầu.

Giao thức được tùy biến có nhiều ứng dụng khác nhau, điển hình là lưu trữ khóa bí mật dùng để bảo đảm tính bí mật và tính toàn vẹn dữ liệu.

Vì vậy, trong phạm vi đồ án này em sẽ đề xuất sử dụng sản phầm sử dụng giao tiếp USB để có thể sản xuất thiết bị nhỏ hơn, giá thành rẻ hơn, cùng với đó là tính mềm dẻo tốt hơn. Với những ưu điểm vừa nêu, thì thiết bị có thể áp dụng cho nhiều sản phẩm tùy biến theo mục đích của doanh nghiệp. Ví dụ có thể tùy biến thiết bị USB như một phần xác thực giao dịch thư điện tử, email, tham gia đầu tư chứng khoán trực tuyến, mua bán hàng hóa online, thanh toán online, chuyển tiền trực tuyến mà không sợ bị mất cắp tiền như với đối với dùng các tài khoản VISA, Master. Bên cạnh đó, còn có thể dùng trong kê khai, nộp thuế trực tuyến với cơ quan hải quan hoặc tương lai gần có thể sử dụng trong chính phủ điện tử.



## 1.2 Mục tiêu cần đạt được

Theo tìm hiểu một số công ty lớn sử dụng USB làm thiết bị CA, em nhận thấy các công ty đang phụ thuộc khá nhiều vào các nhà sản xuất phần cứng (Independent Hardware Vendor - IHV) ví dụ nổi tiếng nhất như SecureMetric.Việc phụ thuộc vào các nhà phát triển phần cứng với cộng với việc đặt số lượng lớn giúp chi phí nghiên cứu và giá thành giảm xuống. Việc này mang lại lợi ích kinh tế trước mắt, tuy nhiên, việc đó cũng làm chúng ta phụ thuộc vào công nghệ của các công ty này. Thay vì việc tùy biến thiết bị theo ý muốn, chúng ta cố gắng ép mình theo khả năng của thiết bị.

Dựa trên các phân tích đánh giá trên, ta nhận thấy với việc sử dụng thiết bị của các nhà cung cấp thiết bị, khả năng tùy biến thiết bị của chúng ta sẽ thấp, và nếu tiếp tục công nghệ nhúng của chúng ta sẽ khó phát triển mạnh hơn nếu không làm chủ được công nghệ. Với ứng dụng được tùy biến từ những phần nhỏ nhất, từ thiết bị đến firmware và ứng dụng giao tiếp, việc tùy biến thiết bị sau này sẽ đơn giản hơn, và phù hợp hơn với nhu cầu của từng bài toán cụ thể.

Mục tiêu được đặt ra ở đây là cần làm chủ công nghệ, từ thiết bị, firmware và ứng dụng. Trên cơ sở đó, tạo nên ứng dụng dùng để làm thiết bị USB Token dùng để xác thực chủ thể.

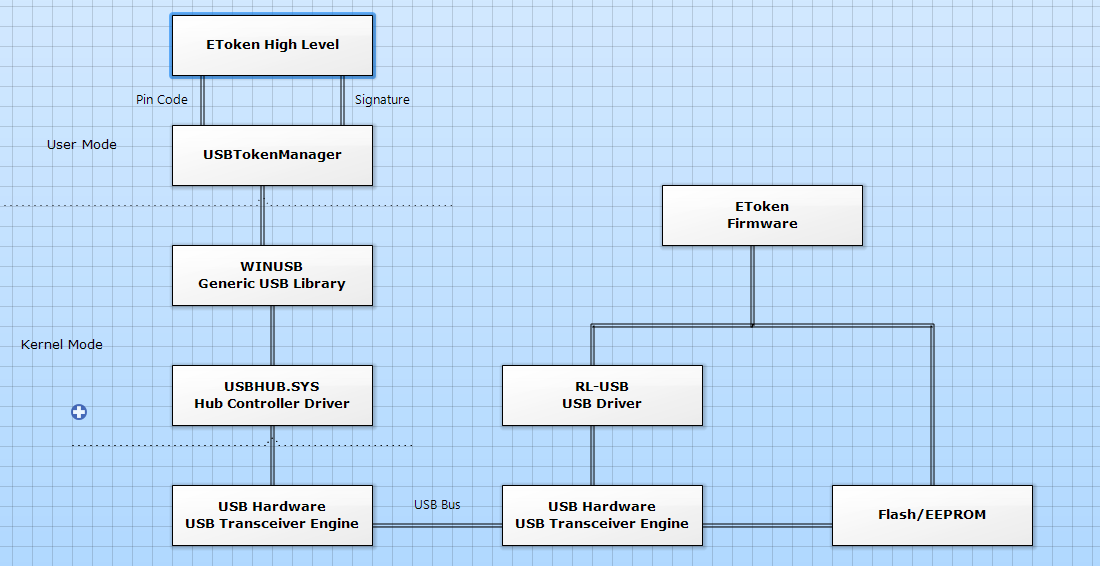
## 1.3 Lựa chọn và định hướng thiết kế

Từ yêu cầu nêu ra bên trên, em tập trung nghiên cứu tìm các thiết bị có giá thành rẻ, kèm theo có EEPROM kèm theo. Từ đó, quyết định chọn vi điều khiển dòng LPC1833 có 16kB bộ nhớ EEPROM dùng để lưu thông tin.

Tiếp theo, chọn kỹ thuật giao tiếp USB với firmware của thiết bị, em chọn sử dụng công nghệ WinUSB trên Windows thay vì sử dụng User-mode framework ( UMDF) hay kernel-mode driver framework (KMDF). Việc lựa chọn này giúp tiết kiệm thời gian phát triển, đồng thời đảm bảo có thể sử dụng đầy đủ tính năng cần cung cấp. Giải thích việc lựa chọn này sẽ được giải thích chi tiết trong Chương 3.

Giao diện ứng dụng trên Windows, sử dụng thư viện Winform trên nền .Net Framework. Việc lựa chọn Winform giúp giao diện ưa nhìn, tiết kiệm thời gian phát triển, và dễ tiếp cận. Ứng dụng trên Windows tương tác với thư việc SQLite cho phép tương tác với cơ sở dữ liệu chữ ký số để sử dụng.

Từ yêu cầu xây dựng thiết bị với giá thành rẻ nhưng vẫn đảm bảo các yếu tố về an toàn và bảo mật, đồ án đề ra giải pháp nghiên cứu và phát triển thiết bị USB Token từ các linh kiện có giá thành rẻ. Ngoài ra xây dựng thuật toán cho phép lưu trữ khóa bí mật trên thiết bị một cách an toàn và khó để có thể tìm ra khóa bí mật nếu thiết bị bị mất. Cùng với đó là ứng dụng trên Windows cho phép giao tiếp với thiết bị, hoạt động trên các máy tính sử dụng Windows. Cuối cùng là giao diện để người dùng sử dụng các chức năng của thiết bị.



*Hình 1.2. Thiết kế kiến trúc USB Etoken*

## 1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày về việc xây dựng, và thiết kế thiết bị, firmware giao tiếp thiết bị. Phần đầu chương giới thiệu, khảo sát các công nghệ nhúng dùng trong thiết bị Token CA hiện có như: thẻ RFID, thẻ từ, thẻ thông minh, thiết bị USB. Tiếp theo, đề xuất giải pháp sẽ sử dụng lưu khóa bí mật: thiết bị USB. Sau đó, khảo sát các linh kiện cần thiết có thể sử dụng để xây dựng thiết bị USB và thiết kế mạch cho thiết bị. Chọn thiết bị phần cứng thử nghiệm để xây dựng thiết bị. Cuối cùng, trên firmware thiết bị, ta thiết kế khuôn dạng gói tin giao tiếp Firmware với PC.

Trong chương 3, em sẽ trình bày việc khảo sát giải pháp giao tiếp giữa thiết bị và ứng dụng trên PC chạy hệ điều hành Windows, sau đó đề xuất giải pháp giao tiếp sử dụng API Windows thông qua các API được export từ driver WinUSB. Tiếp theo là việc xây dựng giao tiếp với firmware thiết bị dựa vào các API vừa nêu trên.

Phần tiếp theo của chương sẽ mô tả các hàm được xây dựng để giao tiếp với firmware vừa nêu. Các hàm này được cung cấp giúp cho bất kỳ một ứng dụng nào sau này có thể tái sử dụng nó để giao tiếp với driver.

Chương 4: Chương này sẽ sử dụng các hàm vừa xây dựng ở chương 3 , ứng dụng giao tiếp với USB tạo USB Token. Ta tìm hiểu khóa bất đối xứng và tiếp theo là ứng dụng của nó trong việc tạo chữ ký số.

Cuối cùng, khảo sát các công nghệ để lập trình giao diện ứng dụng Windows và chọn ra công nghệ phù hợp nhất. Các hình ảnh demo ứng dụng này.

Chương 5: Nêu kết luận về kết quả nghiên cứu, phân tích các vấn đề đã làm được, chưa làm được, các đóng góp nổi bật. Từ đó rút ra bài học kinh nghiệm. Tiếp theo, trình bày hướng phát trển công việc này trong tương lai để hoàn thiện sản phẩm và các ứng dụng khác có thể hoạt động trên các vấn đề đã giải quyết.

**CHƯƠNG III: XÂY DỰNG GIẢI PHÁP VÀ KỊCH BẢN GIAO TIẾP GIỮA FIRMWARE VÀ ỨNG DỤNG TRÊN WINDOWS**

Phần mềm điều khiển trên PC làm nhiệm vụ giao tiếp với phần sụn trên thiết bị thông qua chuđể thực hiện các thao tác trao đổi thông tin. Phần mềm được chia thành hai lớp con:

* Trình điều khiển thiết bị (driver).
* Thư viện API (dll).

## 3.1 Chuẩn USB

USB (Universal Serial Bus – Bus tuần tự đa năng), là một chuẩn được phát triển để kết nối các thiết bị và một thiết bị điều khiển (thường là PC). Mục đích ban đầu của USB là thay thế các loại giao tiếp cũ như cổng tuần tự (serial port) và cổng song song (parallel port). USB kết nối các thiết bị ngoại vi như chuột, bàn phím, máy ảnh số, máy nghe nhạc, ổ đĩa flash và các ổ cứng gắn ngoài. Chính vì sự đa dạng và linh hoạt của USB, ngày nay nó đã được sử dụng trong hầu hết các thiết bị ngoại vi để kết nối với máy tính. USB được chuẩn hóa bởi **USB Implement Forum** (USBIF), tập hợp của các công ty dẫn đầu trong lĩnh vực công nghệ.

### 3.1.1. Tổng quan

USB có thiết kế bất đối xứng, một thiết bị host có nhiều cổng, các thiết bị ngoại vi kết nối đến các cổng đó theo kiến trúc phân tầng. Chỉ có một thiết bị điều khiên bus. Mỗi bus có thể nối tối đa với 127 thiết bị.



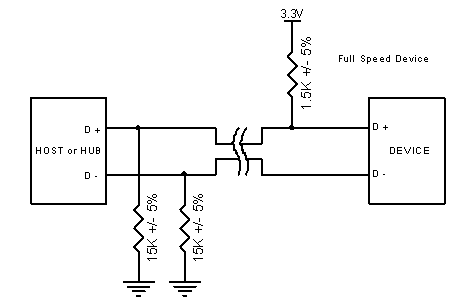
*Hình 3.11. Kết nối USB*

USB có bốn chuẩn tốc độ:

* Super Speed – 4.8Gbits/s
* High Speed – 480Mbits/s
* Full Speed – 12Mbits/s
* Low Speed – 1.5 Mbits/s.

Chuẩn Super Speed đang bắt đầu được thương mại hóa, nhưng số lượng thiết bị hỗ trợ chưa nhiều. Hiện nay chuẩn High Speed là phổ biến nhất. Vi điều khiển thử nghiệm LCP2378 hỗ trợ đến tốc độ FullSpeed, đủ cho việc lưu trữ và trao đổi các đặc trưng của mỗi người.

### 3.1.2. Phần cứng



*Hình 3.12 Kết nối USB*

Kết nối USB có bốn dây. Hai dây cho nguồn điện (Vcc , GND) và hai dây cho tín hiệu vi sai (D+, D-). Thông tin được mã hóa NRZI và truyền đi trên cặp dây D+, D-.

Mức logic là ‘1’ nếu D+ lớn hơn D- 200mV, và ‘0’ nếu D+ nhỏ hơn D- 200mV. Thiết bị FullSpeed và LowSpeed được phân biệt bởi trở kéo trên dây tín hiệu là D+ hay D-. Thiết bị HighSpeed không có trở kéo, ban đầu nó sẽ hoạt động như FullSpeed, sau đó sẽ chuyển sang HighSpeed nếu host hỗ trợ.

Một ưu điểm của USB so với các chuẩn khác là nguồn điện. Thiết bị có thể lấy nguồn trực tiếp từ bus USB mà không cần nguồn điện ngoài. Điện áp trên bus là 5V, dòng cung cấp tối đa là 500mA mỗi cổng.

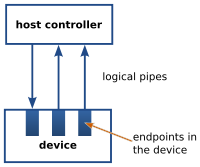
Tần số hoạt động của USB là 48MHz, sai số cho phép với thiết bị HighSpeed là +500ppm. Tương ứng của FullSpeed là 2500ppm và LowSpeed là 15000ppm.

### 3.1.3. Giao thức

Một thiết bị USB vật lý có thể có nhiều thiết bị luân lý. Thí dụ, webcam có thể tích hợp microphone. Đó là các thiết bị phức hợp. Mỗi thiết bị luân lý được USB host gán cho một địa chỉ riêng biệt trên bus và hoạt động như một thiết bị độc lập khác.

Thiết bị USB được hệ điều hành nhận diện và nạp trình điều khiển tương ứng thông qua danh duy nhất gồm định danh nhà sản xuất (Vendor ID) và định danh sản phẩm (Product ID) , do tổ chức USBIF cấp phát.

Các thiết bị luân lý giao tiếp với host thông qua các kênh truyền luân lý gọi là **Pipe.** Pipe kết nối host tới một thực thể luân lý trên thiết bị gọi là EndPoint. Mỗi thiết bị có thể có tới 32 pipe, 16 pipe chiều từ host đến thiết bị và 16 pipe chiều ngược lại.



*Hình 3.13: Đường ống*

Có hai loại pipe: pipe dòng và pipe thông điệp phụ thuộc vào phương pháp truyền. Có bốn phương pháp truyền:

* Isochronous – phương pháp truyền đảm bảo tốc độ nhưng có khả năng mất mát dữ liệu (dữ liệu thời gian thực, hình ảnh, âm thanh).
* Interrupt – phương pháp truyền đảm bảo khả năng đáp ứng, sử dụng cho các thiết bị chủ động (chuột, bàn phím…).
* Bulk –phương pháp truyền một lượng lớn dữ liệu, sử dụng tất cả băng thông còn lại của bus, không có đảm bảo về độ trễ và tốc độ.
* Control – thường để truyền một lượng nhỏ nhữ liệu nhanh, đơn giản, được sử dụng để truyền lệnh và nhận phản hồi .

Pipe dòng là pipe một chiều, kết nối với các endpoint một chiều và truyền theo phương pháp isochronous, interrupt hoặc bulk. Pipe thông điệp là pipe hai chiều và chỉ được sử dụng để điều khiển thiết bị.

Mỗi thiết bị luân lý được gọi là interface, mỗi interface bao gồm một hay nhiều endpoint và thực hiện một chức năng của thiết bị vật lý. Tất cả các thiết bị USB đều có endpoint 0, đó là endpoint đặc biệt không thuộc interface nào, sử dụng để cấu hình thiết bị.

Khi thiết bị USB được kết nối vào USB host, host sẽ bắt đầu quá trình liệt kê (enumration). Host sẽ gửi tín hiệu reset đến thiết bị, tốc độ được xác định qua quá trình reset. Sau đó, host sẽ đọc thông tin về thiết bị và gán cho nó một địa chỉ 7-bit duy nhất. Nếu host hỗ trợ thiết bị này, host sẽ nạp trình điều khiển tương ứng và thiết bị chuyển sang trạng thái đã cấu hình. Nếu usb host khởi động lại, toàn bộ quá trình liệt kê sẽ được thực hiện lại cho tất cả các thiết bị kết nối vào bus.

USB host điều khiển toàn bộ quá trình truyền nhận dữ liệu trên bus, không thiết bị nào có thể tự ý truyền nếu chưa có yêu cầu từ host.

## 3.2. Trình điều khiển thiết bị

Để ứng dụng tầng trên của BioPKI có thể trao đổi dữ liệu với Bio-EToken, hệ điều hành cần phải nạp trình điều khiển (Driver) tương ứng và cung cấp giao diện cho ứng dụng. Do đây là thiết bị tự xây dựng nên ta phải cung cấp trình điều khiển cho hệ điều hành.

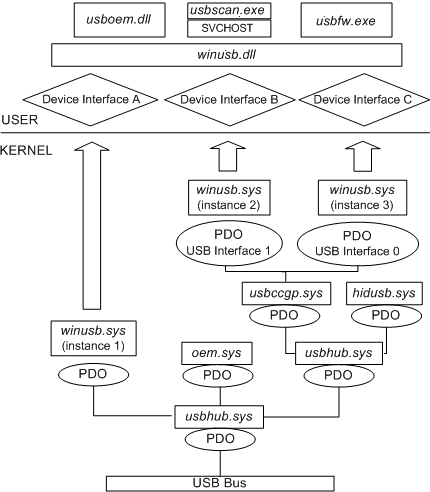
Có hai phương pháp tiếp cập để xây dựng trình điều khiển:

* Xây dựng một trình điều khiển riêng chạy ở nhân hệ điều hành, trình điều khiển sẽ liên lạc với phần cứng usb host và trao đổi thông tin với BioEToken qua bus USB.
* Sử dụng một trình điều khiển chung cho mọi thiết bị USB. Cấu hình lại driver để hỗ trợ BioEToken. Xây dựng thư viện liên kết động chạy ở chế độ người dùng và điều khiển trình điều khiển này trao đổi thông tin với BioEToken.

Phương pháp thứ nhất tương đối phức tạp và thích hợp với các ứng dụng cần tốc độ cao, nhưng dễ sai sót, có thể làm đổ vỡ hệ điều hành. Phương pháp thứ hai đơn giản và dễ triển khai hơn. Chúng tôi đã quyết định chọn phương án thứ hai. Trình điều khiển chung được sử dụng là WinUSB cho các hệ điều hành của Microsoft.

WinUSB hỗ trợ ứng dụng trao đổi dữ liệu với các endpoint của BioEToken, từ đó có thể xây dựng giao thức riêng để truyền thông tin. WinUSB gồm hai thành phần chính:

* Winusb.sys là driver chạy ở nhân hệ điều hành, bên trên các driver điều khiển phần cứng USB.
* Winusb.dll là thư viện liên kết động chạy ở chế độ người dùng, cung cấp giao diện tương tác với Winusb.sys.



*Hình 3.14. Kiến trúc WinUSB*

Công việc còn lại là cấu hình WinUSB để hỗ trợ BioEToken. Các thành phần tối thiểu để cài đặt WinUSB driver gồm có:

* Thư viện WinUSB co-installer: WinUSBCoInstaller.dll.
* Thư viện KMDF (Kernel Mode Driver Framework) co-installer: WdfCoInstallerxxxxx.dll với xxxx là phiên bản của KMDF. Thí dụ WdfCoInstaller01005.dll. WinUSB hoạt động dựa vào vào KMDF do vậy trong bộ cài cần có KMDF.
* File INF mô tả BioEToken.

Từ file INF chuẩn của Microsoft, cần sửa những thông tin sau:

* Trong mục **[MyDevice\_WinUSB.NTx86]** sửa phần VID và PID cho phù hợp với định danh nhà sản xuất và định danh sản phẩm. Trong trường hợp BioEToken đó là 0xAAAA và 0xBBBB.
* Trong mục **[Dev\_AddReg]** sửa GUID bằng GUID của thiết bị. Ở đây GUID cho mỗi thiết bị là duy nhất, có thể sinh trực tuyền hoặc sử dụng công cụ sinh GUID của Microsoft đi kèm với WDF.
* Trong mục **[Strings]** sửa các xâu cho phù hợp với thiết bị.

Nội dung file INF trong trình điều khiển BioEToken

[Version]

Signature = "$Windows NT$"

Class = USB

ClassGUID={36FC9E60-C465-11CF-8056-444553540000}

Provider = %ProviderName%

DriverVer=02/07/2008,1.0.0

;CatalogFile=MyCatFile.cat

; Manufacturer

[Manufacturer]

%ProviderName% = MyDevice\_WinUSB,NTx86,NTamd64

[MyDevice\_WinUSB.NTx86]

%USB\MyDevice.DeviceDesc% =USB\_Install, USB\VID\_AAAA&PID\_BBBB

[MyDevice\_WinUSB.NTamd64]

%USB\MyDevice.DeviceDesc% =USB\_Install, USB\VID\_AAAA&PID\_BBBB

; Installation

[USB\_Install]

Include=winusb.inf

Needs=WINUSB.NT

[USB\_Install.Services]

Include=winusb.inf

AddService=WinUSB,0x00000002,WinUSB\_ServiceInstall

[WinUSB\_ServiceInstall]

DisplayName = %WinUSB\_SvcDesc%

ServiceType = 1

StartType = 3

ErrorControl = 1

ServiceBinary = %12%\WinUSB.sys

[USB\_Install.Wdf]

KmdfService=WINUSB, WinUsb\_Install

UmdfServiceOrder=WINUSB

[WinUSB\_Install]

KmdfLibraryVersion=1.7

[USB\_Install.HW]

AddReg=Dev\_AddReg

[Dev\_AddReg]

HKR,,DeviceInterfaceGUIDs,0x10000,"{6d22450a-fe34-46d2-b019-65dba1ae0aaa}"

[USB\_Install.CoInstallers]

AddReg=CoInstallers\_AddReg

CopyFiles=CoInstallers\_CopyFiles

[CoInstallers\_AddReg]

HKR,,CoInstallers32,0x00010000,"WinUSBCoInstaller.dll","WUDFUpdate\_01007.dll","WdfCoInstaller01007.dll,WdfCoInstaller"

[CoInstallers\_CopyFiles]

WinUSBCoInstaller.dll

WdfCoInstaller01007.dll

WUDFUpdate\_01007.dll

[DestinationDirs]

CoInstallers\_CopyFiles=11

; Source Media

[SourceDisksNames.x86]

1 = %DISK\_NAME%,,,\i386

[SourceDisksNames.amd64]

2 = %DISK\_NAME%,,,\amd64

[SourceDisksFiles.x86]

WinUSBCoInstaller.dll=1

WdfCoInstaller01007.dll=1

WUDFUpdate\_01007.dll=1

[SourceDisksFiles.amd64]

WinUSBCoInstaller.dll=2

WdfCoInstaller01007.dll=2

WUDFUpdate\_01007.dll=2

; Copy Files

[\_CopyFiles\_sys]

winusb.sys

; Destination Directories

[DestinationDirs]

DefaultDestDir = 12 ; %SystemRoot%\system32\drivers

\_CopyFiles\_sys = 12

; Strings

[Strings]

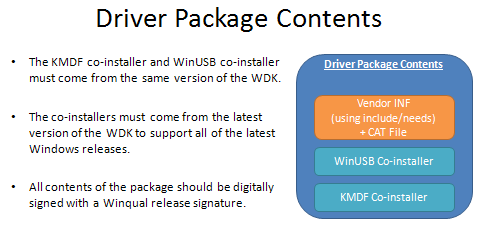
ProviderName="SOICT Research"

USB\MyDevice.DeviceDesc="BIOPKI EToken v2.0"

WinUSB\_SvcDesc="BIOPKI EToken v2.0"

DISK\_NAME="BIOPKI Installation Disc"

Các thành phần trong trình điều khiển BioEToken



*Hình 3.15. WinUSB driver*

## 3.2. Thiết kế giao thức

Mỗi thiết bị EToken sẽ lưu trữ được nhiều khối dữ liệu gọi là đặc trưng sinh trắc khác nhau. Mỗi đặc trưng sinh trắc có thể có kích thước bất kỳ, tối đa 16KB. Các loại đặc trưng sinh trắc mà BioEToken có thể lưu trữ:

* Khóa riêng
* Khóa công khai
* Đặc trưng vân tay
* Đặc trưng giọng nói.
* …

Trước khi PC có thể thực hiện trao đổi dữ liệu, nó cần được xác thực trước. PC sẽ gửi mã PIN đến BioEToken, nếu mã pin đúng, thiết bị sẽ chuyển sang trạng thái sẵn sàng gửi dữ liệu cá nhân về PC. Biểu đồ trao đổi dữ liệu giữa PC và BioEToken

Authentication Ack

**Verify Password**

Authentication Request

**PC Application**

**BioEToken**

Read Request

Data Packet

Data Packet

Data Packet

**Read EEPROM**

*Hình 3.16: Quá trình trao đổi giữa PC và BioEToken*

BioEToken có thể thực hiện được các thao tác sau:

* Xác thực mã pin.
* Đọc một đặc trưng sinh trắc
* Ghi một đặc trưng sinh trắc
* Thiết lập lại mã pin.

BioEToken có nhiều trạng thái, tùy theo số lượng đặc trưng được lưu trữ trên nó.

Quá trình đọc các đặc trưng sinh trắc diễn ra như sau:

* PC gửi mã PIN để thiết bị BioEToken để xác thực.
* BioEToken sẽ tính toán mã PIN và thực hiện đọc từ bộ nhớ, so sánh với mã PIN vừa gửi. Nếu đúng BioEToken sẽ chuyển sang trạng thái sẵn sàng.
* PC gửi yêu cầu đọc đặc trưng đầu tiên. BioEToken sẽ đọc từ EEPROM và gửi trả lại PC.
* PC tiến hành so sánh đặc trưng với dữ liệu sinh trắc sống của người sử dụng.
* Nếu kết quả là khớp, PC sẽ gửi yêu cầu đọc các đặc trưng tiếp theo. BioEToken sẽ tiếp tục đọc từ EEPROM và gửi trả PC.
* …

Quá trình ghi các đặc trưng sinh trắc diễn ra tương tự như đọc. Các đặc trưng sẽ lần lượt được ghi vào bộ nhớ theo thứ tự được ghi.

Quá trình xóa các đặc trưng được thực hiện qua việc thiết lập lại mã PIN. Mỗi khi người dùng thiết lập lại mã PIN, toàn bộ các đặc trưng cũ sẽ bị xóa (vị trí lưu trữ các đặc trưng phụ thuộc vào mã PIN và do đó nó không còn chính xác khi thiết lập PIN mới).

BioEToken và các tầng trên sẽ trao đổi dữ liệu thông qua hai endpoint kiểu bulk: endpoint 1 cho chiều từ thiết bị đến PC và endpoint 2 cho chiều từ PC đến thiết bị. Endpoint 0 là endpoint đặc biệt để liệt kê và khởi tạo thiết bị. Dữ liệu trao đổi trên bus sẽ là các gói tin kích thước 64 byte. Nếu PC gửi một gói tin không hợp ngữ cảnh, BioEToken sẽ bỏ qua và không gửi trả bất kỳ thông tin nào.

**EP1**

**EP2**

**EP0**

**PC**

**BioEToken**

*Hình 3.17. Các Endpoint sử dụng trong BioEToken*

Cấu trúc chung của một gói tin:

|  |  |
| --- | --- |
| Lệnh(1 byte) | Dữ liệu(63 byte) |

Có ba loại thông tin gửi nhận trên bus: thông tin xác thực, thông tin thiết bị và thông tin về đặc trưng. Tương ứng sẽ có ba cấu trúc khác nhau cho mỗi loại thông tin đó.

typedef struct \_DATA\_PACKET // Thông tin về dữ liệu đặc trưng sinh trắc

{

BYTE bCmd; // Mã lệnh

BYTE bIndex; // Đặc trưng cần thao tác

USHORT sDataLen; // Chiều dài đặc trưng

USHORT sDataOffset; // Offset dữ liệu trong đặc trưng

BYTE aData[MAX\_PACKET\_SIZE]; // Dữ liệu đặc trưng

} DATA\_PACKET;

typedef struct \_AUTHENTICATE\_PACKET //Thông tin xác thực

{

BYTE bCmd; // Mã lệnh

BYTE bReserved; // Để dành

BYTE Username[31]; // PIN

BYTE Password[31]; // PIN

} AUTHENTICATE\_PACKET;

typedef struct \_INFO\_PACKET // Thông tin về thiết bị

{

//Number of signature stored in device

BYTE bCmd; // Mã lệnh

BYTE bCount; // Số lượng đặc trưng

USHORT aSignSizes[MAX\_NUMBER\_SIGNATURES];// Kích thước các đặc trưng

} INFO\_PACKET;

## 3.2. Thư viện API

Thư viện API (USBDLL.DLL) sẽ giao tiếp với BioEToken và tầng cao hơn của hệ thống BIOPKI .

Thư viện API cung cấp các hàm sau:

* Xác thực thiết bị.
* Đọc một đặc trưng từ thiết bị.
* Ghi một đặc trưng vào thiết bị.
* Thiết lập lại mã PIN.
* Lấy danh sách các thiết bị.
* Lấy thông tin của một thiết bị.

Để thực hiện được điều đó, thư viện sẽ tương tác với trình điều khiển thiết bị WinUSB đã thiết kế ở trên.

Nguyên mẫu các hàm thư viện được thiết kế như sau:

* Lấy danh sách các thiết bị hiện thời

BOOL WINAPI GetDeviceList(DWORD \*pCount,USBPKI\_DEVICE\_INFO\*pDeviceList);

*Trong đó: - Hàm trả về TRUE nếu thành công.*

*- pCount trả về số thiết bị đang kết nối với hệ thống*

*- pDeviceList là danh sách các thiết bị kết nối với hệ thống.*

*Cấu trúc một phần tử của danh sách như sau*

typedef struct \_USBPKI\_DEVICE\_INFO

{

USHORT iVersion;

WCHAR szDeviceString[MAX\_DEVICE\_STRING\_LENGTH];

} USBPKI\_DEVICE\_INFO;

* Xác thực một thiết bị

BOOL WINAPI AuthenticateDevice(USBPKI\_DEVICE\_INFO\*pDevice,BYTE\* sUsername,BYTE \* sPassword);

*Trong đó: - Hàm trả về TRUE nếu xác thực thành công.*

*- pDevice là con trỏ đến một thiết bị kết nối tới hệ thống.*

*- sUsername và sPassword là tên và mật khẩu.*

* Lấy thông tin về một thiết bị

BOOL WINAPI GetDeviceInformation(IN USBPKI\_DEVICE\_INFO\*pDevice,OUT USHORT\* iNumSigns,OUT USHORT \* aSignSizes);

*Trong đó: - Hàm trả về TRUE nếuthông tin lấy thành công.*

*- pDevice là con trỏ đến một thiết bị cần lấy thông tin*

*- iNumSigns và aSignSizes là số lượng đặc trưng và kích thước các đặc trưng lưu trên thiết bị đó .*

* Đọc một đặc trưng sinh trắc

BOOL WINAPI ReadSignature(IN USBPKI\_DEVICE\_INFO\*pDevice,IN BYTE iIndex,OUT BIO\_SIGNATURE \* pSignature,HWND hWnd = NULL);

*Trong đó: - Hàm trả về TRUE nếu thành công*

*- pDevice là con trỏ đến thiết bị cần đọc*

*- iIndex là vị trí đặc trưng cần đọc*

*- pSignature là kết quả đặc trưng đọc được*

*- hWnd là handle cửa sổ nhận thông tin về quá trình đọc*

*Lưu ý mỗi đặc trưng lưu trong cấu trúc như sau*

typedef struct \_BIO\_SIGNATURE

{

DWORD iLen;

BYTE SignData[MAX\_SIGNATURE\_SIZE];

}BIO\_SIGNATURE;

* Ghi một đặc trưng sinh trắc

BOOL WINAPI WriteSignature(IN USBPKI\_DEVICE\_INFO\*pDevice,IN BYTE iIndex,IN BIO\_SIGNATURE \* pSignature,HWND hWnd = NULL);

*Trong đó: - Hàm trả về TRUE nếu thành công*

*- pDevice là con trỏ đến thiết bị cần đọc.*

*- iIndex là vị trí đặc trưng cần ghi.*

*- pSignature là nội dung đặc trưng muốn ghi*

*- hWnd là handle cửa sổ nhận thông tin về quá trình ghi.*

* Thiết lập lại mã PIN

BOOL WINAPI SetPasswordDevice(USBPKI\_DEVICE\_INFO\*pDevice,BYTE\* user,BYTE \* pass);

*Trong đó: - Hàm trả về TRUE nếu thiết lập thành công*

*- pDevice là con trỏ đến một thiết bị kết nối tới hệ thống.*

*- sUsername và sPassword là tên và mật khẩu.*

Để thực hiện được các hàm trên , USBDLL cần sử dụng các hàm sau của WinUSB.

* Mở một thiết bị

hDev = CreateFile(devicePath,

GENERIC\_WRITE | GENERIC\_READ,

FILE\_SHARE\_WRITE | FILE\_SHARE\_READ,

NULL,

OPEN\_EXISTING,

FILE\_ATTRIBUTE\_NORMAL | FILE\_FLAG\_OVERLAPPED,

NULL);

Với hDev là handle của thiết bị, devicePath là đường dẫn tới thiết bị

* Khởi tạo handle của WinUSB

bResult = WinUsb\_Initialize(deviceHandle, &usbHandle);

với usbHandle là handle của WinUSB

* Đọc dữ liệu từ một PIPE

bResult = WinUsb\_ReadPipe(usbHandle,

iPipeID,

szBuffer,

24,

&bytesRead,

NULL);

Với usbHandle là handle có được từ bước 2, iPipeID là số nguyên chỉ PIPE IN của BioEToken, trong trường hợp này là 0x81

* Ghi dữ liệu ra PIPE

bResult = WinUsb\_WritePipe(usbHandle,

iPipeID,

szBuffer,

24,

&bytesWritten,

NULL);

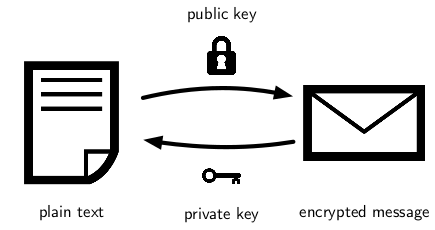
Với usbHandle là WinUSB handle có từ bước 2, iPipeID là số nguyên chỉ PIPE OUT của BioEToken, trong trường hợp này là 0x02.

**CHƯƠNG IV: ỨNG DỤNG TẠO PHẦN MỀM CHỮ KÝ SỐ**

## Cơ sở lý thuyết

### Tìm hiểu mật mã hóa khóa công khai

Mật mã hóa khóa công khai hay còn được gọi với một cái tên khác là mã hóa khóa bất đối xứng (Public Key Cryptography), nó được thiết kế sao cho khóa sử dụng trong quá trình mã hóa khác biệt với khóa được sử dụng trong quá trình giải mã. Hơn thế nữa, khóa dùng trong quá trình giải mã không thể được tính toán hay suy luận từ khóa dùng để mã hóa và ngược lại, tức là hai khóa này có quan hệ với nhau về mặt toán học nhưng không thể suy diễn được ra nhau. Thuật toán này được gọi là Public-Key bởi vì khóa dùng cho việc mã hóa được công khai cho tất cả mọi người. Một người hòan toàn xa lạ có thể dùng khóa này để mã hóa dữ liệu nhưng chỉ duy nhất người mà có khóa giải mã tương ứng mới có thể đọc được dữ liệu mà thôi. Do đó trong thuật mã hóa này Encryption key được gọi là Public key còn Decryption Key được gọi là Private key.



Tại sao lại có sự xuất hiện của Mã hóa khóa công khai? Như đã trình bày với các bạn ở trên, thuật toán mã hóa khóa riêng tuy có tốc độ thực hiện rất nhanh nhưng nó có một số nhược điểm như sau:

* Khóa phải được trao đổi theo một kênh bí mật.
* Nếu mất khóa thì thông tin hoàn toàn có thể bị lấy cắp hoặc giả mạo.
* Khóa cần phải thay đổi một cách định kì
* Khi số lượng người dùng tăng lên thí số lượng khóa được sử dụng cũng tăng lên.

Vậy là điểm yếu của thuật toán mã hóa khóa riêng nằm ở khâu quản lý khóa thế nào cho hợp lý. Mã hóa khóa công khai ra đời đã giải quyết được vấn đề này. Hình minh họa ở trên cho chúng ta thấy được quá trình truyền tin an toàn dựa vào hệ thống mã hóa khóa công khai. Như các bạn thấy, trong hệ thống mã hóa này thì mỗi một người sử dụng khi tham gia vào đều được cấp 2 khóa : Một khóa dùng cho việc mã hóa dữ liệu (Public key) và một khóa dùng cho việc giải mã dữ liệu (Private key), trong đó Public key được đưa ra cho tất cả mọi người cùng biết, còn Private key phải được giữ kín một cách tuyệt đối.Giả sử hai phía muốn truyền tin cho nhau thì quá trình truyền sử dụng mã hóa khóa công khai được thực hiện như sau :

Sender yêu cầu cung cấp hoặc tự tìm khoá công khai của Receiver trên một Server chịu trách nhiệm quản lý khoá công khai.

Sau đó hai phía thống nhất thuật toán dùng để mã hóa dữ liệu, Sender sử dụng khóa công khai của Receiver cùng với thuật toán đã thống nhất để mã hóa thông tin bí mật.

Thông tin sau khi mã hóa được gửi tới Receiver, lúc này chính Sender cũng không thể nào giải mã được thông tin mà anh ta đã mã hóa (khác với mã hóa khóa riêng).

Khi nhận được thông tin đã mã hóa, Receiver sẽ sử dụng khóa bí mật của mình để giải mã và lấy ra thông tin ban đầu.

Vậy là với sự ra đời của Mã hóa khóa công khai thì khóa được quản lý một cách linh hoạt và hiệu quả hơn.Người sử dụng chỉ cần bảo vệ khóa Private key. Hệ thống này an toàn hơn nhiều so với mã hóa khóa riêng, người mã hóa không thể giải mã được dữ liệu đã mã hóa bằng khóa công khai của người khác.Tuy nhiên nhược điểm của Mã hóa khóa công khai nằm ở tốc độ thực hiện, nó chậm hơn mã hóa khóa riêng cỡ ~1000 lần. Do đó người ta thường kết hợp hai hệ thống mã hóa khóa riêng và công khai lại với nhau và được gọi là Hybrid Cryptosystems (Hệ thống mã hóa lai). Một số thuật toán mã hóa công khai phổ biến :

* RSA : Cái tên RSA là ba chữ cái bắt đầu từ ba cái tên của ba tác giả: Ron Rivest, Adi Shamir ,Len Adleman. Sử dụng đồng thời cho mã hóa khóa công khai và chữ kí điện tử. Độ an toàn của thuật toán mã hóa RSA dựa trên việc phân tích một số nguyên tố rất lớn thành hai số nguyên tố.

### Thuật toán RSA

Trong mật mã học, RSA là một thuật toán mật mã hóa khóa công khai. Đây là thuật toán đầu tiên phù hợp với việc tạo ra chữ ký điện tử đồng thời với việc mã hóa. Nó đánh dấu một sự tiến bộ vượt bậc của lĩnh vực mật mã học trong việc sử dụng khóa công cộng. RSA đang được sử dụng phổ biến trong thương mại điện tử và được cho là đảm bảo an toàn với điều kiện độ dài khóa đủ lớn.

Thuật toán RSA có hai khóa: khóa công khai (hay khóa công cộng) và khóa bí mật (hay khóa cá nhân). Mỗi khóa là những số cố định sử dụng trong quá trình mã hóa và giải mã. Khóa công khai được công bố rộng rãi cho mọi người và được dùng để mã hóa. Những thông tin được mã hóa bằng khóa công khai chỉ có thể được giải mã bằng khóa bí mật tương ứng. Nói cách khác, mọi người đều có thể mã hóa nhưng chỉ có người biết khóa cá nhân (bí mật) mới có thể giải mã được.

### Sự cần thiết của chữ ký số

Cơ chế xác thực thông tin bảo vệ hai bên trao đổi thông điệp khỏi bên thứ ba. Tuy nhiên cơ chế này không bảo vệ được một bên khi bên kia cố ý vi phạm, ví dụ như:

* A có thể làm giả một thông điệp và tuyên bố rằng thông điệp này do B gửi. A chỉ cần tạo một thông điệp và thêm vào mã xác thực sử dụng khóa mà A và B chia sẻ.
* B có thể phủ nhận đã gửi thông điệp. Bởi vì A có thể giả mạo thông điệp nên không có cách nào chứng minh sự thật là B đã gửi.

Trong trường hợp không có sự tin tưởng hoàn toàn giữa người gửi và người nhận, cần có một cơ chế tốt hơn xác thực. Giải pháp tốt nhất cho vấn đề này là chữ ký số.

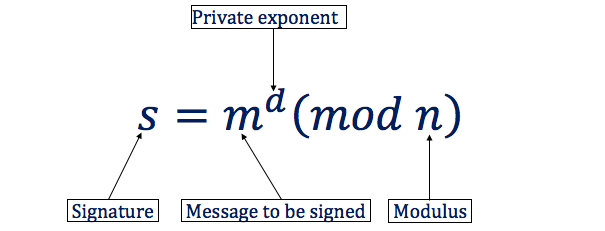
Dựa trên nền tảng các tính chất trên, chúng ta có thể hình thành các yêu cầu cho chữ ký số như sau:

* Chữ ký số là một số các bit phụ thuộc vào thông điệp được ký được đính kèm với thông điệp.
* Chữ ký số phải sử dụng những thông tin độc lập với người gửi để ngăn chặn giả mạo.
* Phải tương đối dễ dàng để tạo ra chữ ký số
* Phải tương đối dễ dàng để nhận ra và xác minh chữ ký số.
* Không thể giả mạo được một chữ ký số, tức là tạo một thông điệp mới cho một chữ ký đã tồn tại hoặc tạo một chữ ký giả cho một thông điệp cho trước.
* Có thể sao chép được chữ ký số để lưu trữ.

## Quy trình ký file bằng chữ ký số

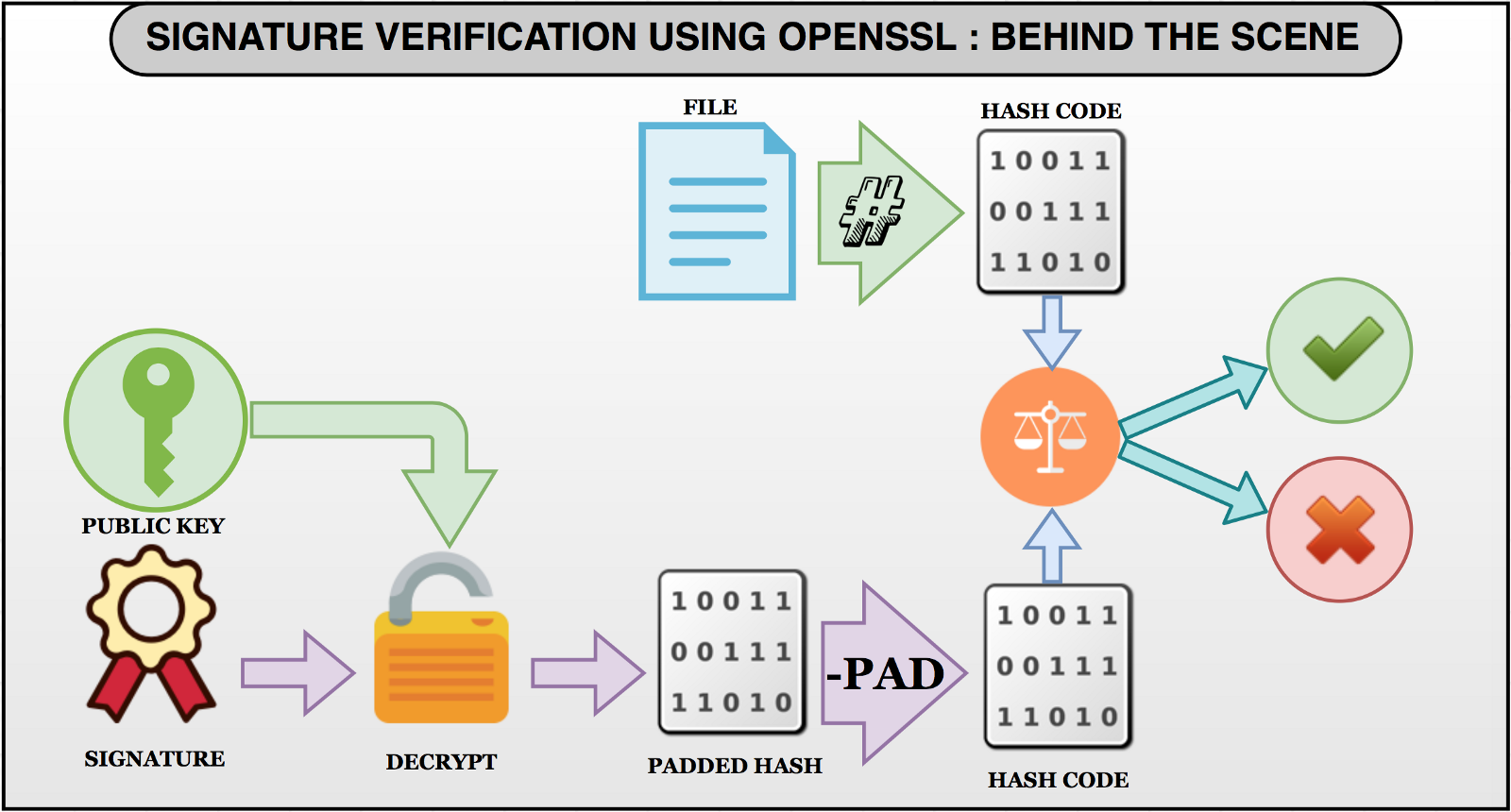
Quy trình ký dữ liệu bằng chữ ký số diễn ra như sau:

* Bước 1: Băm dữ liệu
  + Tin nhắn (dữ liệu) đi qua một hàm mật mã để tạo ra một mảng băm của dữ liệu.
  + SHA1 tạo ra mảng băm 160 bit (20 byte).
  + SHA224, SHA256, SHA384, SHA512, MD4, MD5 là một vài thuật toán băm thông điệp khác có sẵn trong openssl.
* Bước 2: Thêm đệm cho giá trị băm
  + Giá trị băm (20 byte trong trường hợp SHA1) được mở rộng đến kích thước RSA key bằng cách thêm đệm tiền tố vào trước mã băm.
  + Sơ đồ đệm mặc định trong openssl là PKCS1.
  + Sơ đồ đệm PKCS # 1v1.5 cho SHA1:
  + Sơ đồ đệm PKCS1 cho thuật toán phân loại SHA1
* Bước 3: Lấy modulus n và số mũ riêng từ khóa bí mật
  + Dữ liệu modulus n và số mũ riêng được lấy từ thiết bị đã được lưu vào thiết bị trước đó. Việc lấy dữ liệu này được lấy theo
* Bước 4: Ký mã băm với modulus n và số mũ riêng



Hình 4.2 Mã hóa dữ liệu với Modulus n và số mũ riêng

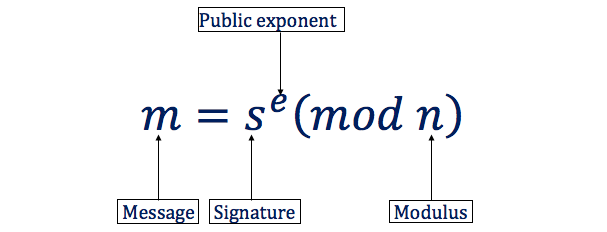
* + Mã hóa tin nhắn với mô đun và số mũ riêng để có chữ ký



Hình 4.1 Quy trình tạo, sử dụng chữ ký số

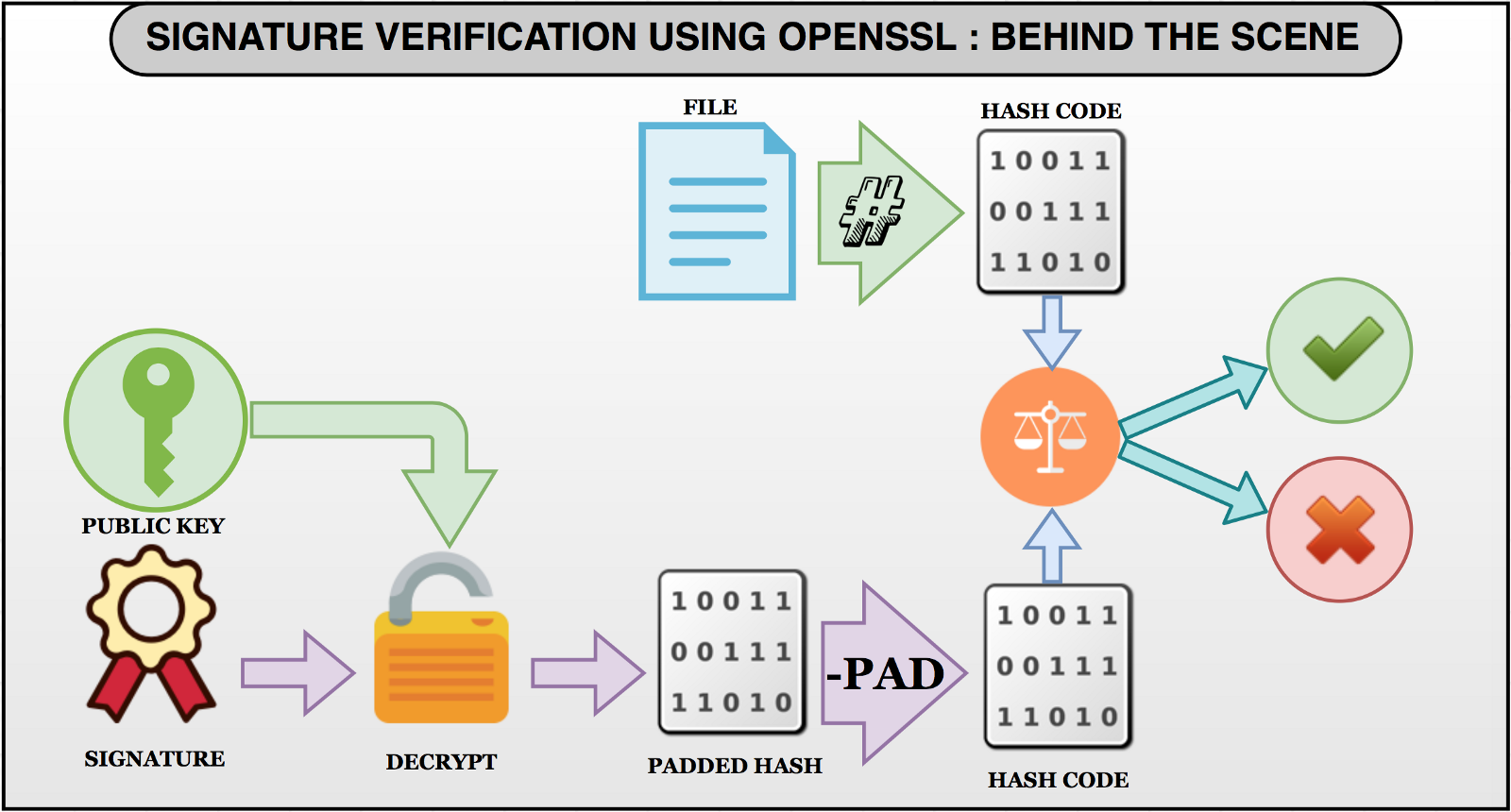
## Quy trình xác thực chữ ký số

Quy trình xác thực chữ ký số được diễn ra như sau:

* Bước 1:Lấy thông tin modulus n và số mũ công khai từ khóa công khai
  + Các nội dung của khóa công khai: Public key chứa Modulus n, số mũ công khai và kích thước khóa.
  + Khóa công khai được cung cấp từ server lưu trữ database tương ứng với chủ thể được cung cấp.
* Bước 2: Chuyển đổi chữ ký sang dạng mã băm đã được chèn thêm padding

Hình 4.3: Giải mã chữ ký bằng cách sử dụng modulus n và số mũ công khai để nhận được gói tin

* Bước 3: Xóa phần padding để có được mã băm của gói tin
* Bước 4: So sánh mã băm vừa được giải mã từ chữ ký số. So sánh mã băm này với mã băm của gói tin để kết luận tính toàn vẹn của dữ liệu.



Hình 4.1 Quy trình xác minh chữ ký số

**CHƯƠNG V: KẾT LUẬN**

**DANH MỤC TÀI LIỆU THAM KHẢO**

[1] Giới thiệu về ASP.NET MVC <https://www.asp.net/>, truy cập tháng 5/2018

[2] Giới thiệu về SQL Server <https://en.wikipedia.org/wiki/Microsoft_SQL_Server>, truy cập tháng 5/2018