|  |
| --- |
| BAN CƠ YẾU CHÍNH PHỦ  **HỌC VIỆN KỸ THUẬT MẬT MÃ**  ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯  Logo HvKTMM |
| ĐỒ ÁN TỐT NGHIỆP  **XÂY DỰNG ỨNG DỤNG KÝ SỐ, MÃ MẬT FILE PDF TRÊN WEB**  Ngành: An toàn thông tin  Mã số: 7.48.02.02  *Sinh viên thực hiện*:  **Phạm Đức Tiến**  Lớp: AT14H  *Người hướng dẫn*:  **TS. Lê Quang Huy**  Cục chứng thực số & bảo mật thông tin  Hà Nội, 2022 |
| BAN CƠ YẾU CHÍNH PHỦ  **HỌC VIỆN KỸ THUẬT MẬT MÃ**  ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯  Logo HvKTMM |
| ĐỒ ÁN TỐT NGHIỆP  **XÂY DỰNG ỨNG DỤNG KÝ SỐ, MÃ MẬT FILE PDF TRÊN WEB**  Ngành: An toàn thông tin  Mã số: 7.48.02.02  *Sinh viên thực hiện*:  **Phạm Đức Tiến**  Lớp: AT14H  *Người hướng dẫn*:  **TS. Lê Quang Huy**  Cục chứng thực số & bảo mật thông tin  Hà Nội, 2022 |

# **LỜI CẢM ƠN**

Tôi xin cảm ơn các Thầy, Cô giáo ở Khoa An toàn thông tin, Phòng Đào tạo, Học viện Kỹ Thuật Mật Mã đã giảng dạy và truyền thụ cho tôi những kiến thức quý báu trong suốt thời gian tôi học tập và nghiên cứu tại trường.

Tôi xin gửi lời cảm ơn chân thành nhất tới TS. Lê Quang Huy, người thầy đã cho tôi những định hướng, tận tình chỉ bảo giúp đờ, cho tôi nhừng ý kiến rất quý báu đê tôi hoàn thành đồ án tốt nghiệp này.

Tôi cũng xin tỏ lòng biết ơn về sự động viên giúp đỡ của gia đình và bạn bè, đồng nghiệp, nhừng người luôn quan tâm sát cánh bên tôi và là nguồn động viên khích lệ, tạo cho tôi có được những điều kiện tốt nhất trong suốt quá trình học tập của mình.

Tôi xin chân thành cảm ơn!

Hà Nội, 10 tháng 6 năm 2022

**Sinh viên**

**Phạm Đức Tiến**

# **LỜI CAM ĐOAN**

Tôi xin cam đoan bản đồ án này do tôi tự nghiên cứu dưới sự hướng dẫn của thầy giáo TS. Lê Quang Huy.

Để hoàn thành đồ án này, tôi chỉ sử dụng những tài liệu đã ghi trong mục tài liệu tham khảo, ngoài ra không sử dụng bất cứ tài liệu nào khác mà không được ghi.

Nếu sai, tôi xin chịu mọi hình thức kỷ luật theo quy định của Học viện.

Hà Nội, ngày 10 tháng 6 năm 2022

**Sinh viên thực hiện**

(Ký tên và ghi rõ họ tên)

**Phạm Đức Tiến**

MỤC LỤC

[LỜI CẢM ƠN iii](#_Toc106190864)

[LỜI CAM ĐOAN iv](#_Toc106190865)

[DANH MỤC HÌNH VẼ vii](#_Toc106190866)

[LỜI CẢM ƠN x](#_Toc106190867)

[VIẾT TẮT xi](#_Toc106190868)

[MỞ ĐẦU 1](#_Toc106190869)

[CHƯƠNG 1: HỆ THỐNG CHỨNG THỰC SỐ 3](#_Toc106190870)

[1.1 Mật mã khóa công khai 3](#_Toc106190871)

[***1.1.1*** ***Khái niệm*** 3](#_Toc106190872)

[***1.1.2*** ***Các thuật toán sử dung*** 4](#_Toc106190873)

[1.1.2.1 Thuật toán RSA 4](#_Toc106190874)

[1.1.2.2 Phương thức trao đổi khóa Diffie-Hellman 5](#_Toc106190875)

[1.2 Chứng thực số 5](#_Toc106190876)

[***1.2.1*** ***Một số khái niệm*** 5](#_Toc106190877)

[1.2.1.1 Cơ sở hạ tầng khóa công khai 5](#_Toc106190878)

[1.2.1.2 Chứng thực số 6](#_Toc106190879)

[1.2.1.3 Chứng thư số 6](#_Toc106190880)

[1.3 Các thành phần của hệ thống chứng thực số 10](#_Toc106190881)

[***1.3.1*** ***Thầm quyền chứng thực (CA- Certification Authority)*** 12](#_Toc106190882)

[***1.3.2*** ***Thẩm quyền đăng ký (RA- Registration Authority)*** 13](#_Toc106190883)

[***1.3.3*** ***Thẩm quyền xác thực (VA- Validation Authority)*** 15](#_Toc106190884)

[***1.3.4*** ***Thực thể cuối (EE- End Entity)*** 15](#_Toc106190885)

[***1.3.5*** ***Kho chứng thư số*** 16](#_Toc106190886)

[***1.3.6*** ***Hủy bỏ chứng thư số*** 16](#_Toc106190887)

[***1.3.7*** ***Sao lưu và khôi phục khóa*** 17](#_Toc106190888)

[***1.3.8*** ***Cập nhật khóa tự động*** 17](#_Toc106190889)

[***1.3.9*** ***Lịch sử khóa*** 17](#_Toc106190890)

[1.4 Các mô hình kiến trúc tin cậy 18](#_Toc106190891)

[***1.4.1*** ***Mô hình phân cấp*** 18](#_Toc106190892)

[***1.4.2*** ***Mô hình mạng lưới*** 20](#_Toc106190893)

[***1.4.3*** ***Mô hình danh sách tin cậy*** 20](#_Toc106190894)

[***1.4.4*** ***Mô hình CA đơn*** 21](#_Toc106190895)

[***1.4.5*** ***Mô hình cầu nối*** 22](#_Toc106190896)

[***1.4.6*** ***Chứng thực chéo*** 23](#_Toc106190897)

[1.5 Các hoạt động chính của hệ thống chứng thực số 24](#_Toc106190898)

[***1.5.1*** ***Chứng thực cặp khóa*** 24](#_Toc106190899)

[***1.5.2*** ***Khẳng định tính hợp lệ, tin cậy*** 25](#_Toc106190900)

[***1.5.3*** ***Dịch vụ cấp dấu thời gian*** 26](#_Toc106190901)

[1.6 Ứng dụng ký số 28](#_Toc106190902)

[1.7 Ứng dụng mã mật 29](#_Toc106190903)

[1.8 Ứng dụng xác thực chống chối bỏ 30](#_Toc106190904)

[CHƯƠNG 2: Công nghệ ký số, mã mật file PDF và xây dựng ứng dụng web 35](#_Toc106190905)

[2.1. Ký số, mã mật file PDF 35](#_Toc106190906)

[***2.1.1.*** ***Giới thiệu tổng quan về PDF*** 35](#_Toc106190907)

[2.1.1.1. Lịch sử 35](#_Toc106190908)

[2.1.1.2. Định nghĩa 35](#_Toc106190909)

[***2.1.2.*** ***Cấu trúc file PDF*** 36](#_Toc106190910)

[***2.1.3.*** ***Chữ ký số trong file PDF*** 40](#_Toc106190911)

[2.1.3.1. Chữ ký số 40](#_Toc106190912)

[2.1.3.2. Tạo và kiểm tra chữ ký số 49](#_Toc106190913)

[2.1.3.3. Chuẩn chữ ký số file PDF ISO 32000-1 và PadES 53](#_Toc106190914)

[2.1.3.4. Các thuật toán chữ ký số thông dụng 55](#_Toc106190915)

[2.1.3.5. Các thuật toán hàm băm mật mã 58](#_Toc106190916)

[***2.1.4.*** ***Mã mật file PDF*** 59](#_Toc106190917)

[2.1.4.1. Giới thiệu về mã mật file PDF 59](#_Toc106190918)

[2.1.4.2. Từ điển mã hóa khóa công khai 62](#_Toc106190919)

[2.1.4.3. Mã mật hóa và giải mã mật 64](#_Toc106190920)

[2.1.4.4. Trình xử lý bảo mật khóa công khai 66](#_Toc106190921)

[2.2. Công nghệ xây dựng ứng dụng web 67](#_Toc106190922)

[***2.2.1.*** ***Tổng quan về ứng dụng web*** 67](#_Toc106190923)

[***2.2.2.*** ***Công nghệ xây dựng ứng dụng Web*** 67](#_Toc106190924)

[2.2.2.1. ASP.NET 67](#_Toc106190925)

[2.2.2.2. Angular 70](#_Toc106190926)

[***2.2.3.*** ***Ký số, mã mật file pdf sử dụng Docotic.Pdf library*** 71](#_Toc106190927)

[3. XÂY DỰNG ỨNG DỤNG WEB KÝ SỐ, MÃ MẬT FILE PDF 75](#_Toc106190928)

[3.1. Tổng quan 75](#_Toc106190929)

[3.2. Phân tích yêu cầu và thiết kế ứng dụng web 75](#_Toc106190930)

[***3.2.1.*** ***Phân tích yêu cầu*** 75](#_Toc106190931)

[***3.2.2.*** ***Thiết kế ứng dụng web*** 75](#_Toc106190932)

[3.2.2.1. Ký số 75](#_Toc106190933)

[3.2.2.2. Xác thực chữ ký số 76](#_Toc106190934)

[3.2.2.3. Mã mật 77](#_Toc106190935)

[3.2.2.4. Giải mã 78](#_Toc106190936)

[**3.3.** **Môi trường hệ thống** 79](#_Toc106190937)

[***3.3.1.*** ***Các thành phần của hệ thống*** 79](#_Toc106190938)

[***3.3.2.*** ***Các chức năng của hệ thống*** 79](#_Toc106190939)

[3.4. Xây dựng website và thử nghiệm 80](#_Toc106190940)

[*3.4.1.* *Xây dựng website* 80](#_Toc106190941)

[3.4.2. Thử nghiệm 82](#_Toc106190942)

[KẾT LUẬN 83](#_Toc106190943)

[DANH MỤC TÀI LIỆU THAM KHẢO 84](#_Toc106190944)

# **DANH MỤC HÌNH VẼ**

[Hình 1.1. Mật mã khóa công khai 3](#_Toc106533637)

[Hình 1.2. Ví dụ về chứng thư số 7](#_Toc106533638)

[Hình 1.3. Cấu trúc chung chứng thư số X.509 8](#_Toc106533639)

[Hình 1.4. Mô hình hoạt động của hệ thống CTS 11](#_Toc106533640)

[Hình 1.5. Các thành phần trong hệ thống PKI 12](#_Toc106533641)

[Hình 1.6. Quá trình xác thực dựa trên CA 13](#_Toc106533642)

[Hình 1.7. Mô hình phân cấp 19](#_Toc106533643)

[Hình 1.8. Mô hình mạng lưới 20](#_Toc106533644)

[Hình 1.9. Mô hình danh sách tin cậy 21](#_Toc106533645)

[Hình 1.10. Mô hình CA đơn 22](#_Toc106533646)

[Hình 1.11. Mô hình cầu nối 23](#_Toc106533647)

[Hình 1.12. Mô hình chứng thực chéo 24](#_Toc106533648)

[Hình 1.13. Sơ đồ chứng thực cặp khóa 25](#_Toc106533649)

[Hình 1.14. Sơ đồ cấp dấu thời gian 27](#_Toc106533650)

[Hình 1.15. Mô hình tạo dấu thời gian an toàn 28](#_Toc106533651)

[Hình 1.16. Kiểm tra tính hợp lệ của dấu thời gian an toàn 28](#_Toc106533652)

[Hình 1.17. Ứng dụng ký số 29](#_Toc106533653)

[Hình 1.18. Ứng dụng mã mật 30](#_Toc106533654)

[Hình 1.19. Ứng dụng xác thực 31](#_Toc106533655)

[Hình 1.20. Xác thực từ xa dựa trên khoá công khai 32](#_Toc106533656)

[Hình 1.21. Xác thực dựa trên bên thứ 3 tin cậy 32](file:///D:\final.finalyyy.docx#_Toc106533657)

[Hình 1.22. Ứng dụng chống chối bỏ 33](#_Toc106533658)

[Hình 2.1. Cấu trúc cơ bản của tệp PDF 37](#_Toc106533659)

[Hình 2.2. Chi tiết về cấu trúc file PDF 38](#_Toc106533660)

[Hình 2.3. Cấu trúc PDF sau khi được cập nhật gia tăng 39](#_Toc106533661)

[Hình 2.4. Ví dụ về tệp PDF 40](#_Toc106533662)

[Hình 2.5. Từ điển chữ ký 41](#_Toc106533663)

[Hình 2.6. Đa chữ ký và cập nhật gia tăng 49](#_Toc106533664)

[Hình 2.7. Giá trị ByteRange và chữ ký 50](#_Toc106533665)

[Hình 2.8. Quá trình ký số 50](#_Toc106533666)

[Hình 2.9. Giá trị ByteRange và chữ ký 52](#_Toc106533667)

[Hình 2.10. Quá trình xác thực chữ ký số 52](#_Toc106533668)

[Hình 2.11. Cấu trúc file PDF trước và sau mã hóa 59](#_Toc106533669)

[Hình 2.12. Public-key encryption algorithm 65](#_Toc106533670)

[Hình 2.13. ASP.Net 68](#_Toc106533671)

[Hình 2.14. Angular 71](#_Toc106533672)

[Hình 2.15. ký số sử dụng thư viện Docotic.Pdf 72](#_Toc106533673)

[Hình 2.16. mã hóa file PDF sử dụng thư viện Docotic 72](#_Toc106533674)

[Hình 2.17 Xác thực chữ ký số. 74](#_Toc106533675)

[Hình 3.1. Sơ đồ chức năng ký số 76](#_Toc106533676)

[Hình 3.2. Sơ đồ xác thực ký số 77](#_Toc106533677)

[Hình 3.3. sơ đồ chức năng mã hóa 78](#_Toc106533678)

[Hình 3.4. sơ đồ chức năng giải mã 79](#_Toc106533679)

[Hình 3.5. Chức năng chính của user 80](#_Toc106533680)

[Hình 3.6. giao diện trang chủ 80](#_Toc106533681)

[Hình 3.7. Giao diện chức năng ký số 81](#_Toc106533682)

[Hình 3.8. giao diện chức năng xác thực chữ ký 81](#_Toc106533683)

[Hình 3.9. giao diện chức năng mã hóa 81](#_Toc106533684)

[Hình 3.10. giao diện chức năng giải mã 82](#_Toc106533685)

[Hình 3.11. File sau khi ký số 82](#_Toc106533686)

[Hình 3.12. Xác thực chữ ký số 83](#_Toc106533687)

[Hình 3.13. File đã được mã hóa và mở bằng trình duyệt thông thường 83](#_Toc106533688)

[Hình 3.14. File đã mã hóa và mở bằng phần mềm Adobe Acrobat 84](#_Toc106533689)

[Hình 3.15. File đã được giải mã 84](#_Toc106533690)

**DANH MỤC BẢNG BIỂU**

[Bảng 2.1. Các mục nhập của từ điển chữ ký 47](#_Toc106190162)

[Bảng 2.2. Các thuật toán và độ dài khóa tối đa được hỗ trợ bởi các bảng mã và phiên bản PDF khác nhau. 48](#_Toc106190163)

[Bảng 2.3. Các giá trị của V 62](#_Toc106190164)

[Bảng 2.4. Các mục nhập từ điển mã hóa bổ sung cho các trình xử lý bảo mật khóa công khai 63](#_Toc106190165)

[Bảng 2.5. Quyền truy cập của người dùng trình xử lý bảo mật khóa công khai 64](#_Toc106190166)

[Bảng 2.6. Ngoài ra còn hỗ trợ các thuật toán mã hóa khác nhau 73](#_Toc106190167)

[Bảng 2.7. các tham số về thuât toán mã hóa 74](#_Toc106190168)

# **LỜI CẢM ƠN**

Tôi xin cảm ơn các Thầy, Cô giáo ở Khoa An toàn thông tin, Phòng Đào tạo, Học viện Kỹ thuật mật mã đã giảng dạy và truyền thụ cho tôi những kiến thức quý báu trong suốt thời gian tôi học tập và nghiên cứu tại trường.

Tôi xin gửi lời cám ơn chân thành nhất tới TS. Lê Quang Huy, người thầy đã cho tôi những định hướng, tận tình chỉ bảo giúp đỡ, cho tôi những ý kiến rất quý báu để tôi hoàn thành đồ án tốt nghiệp này.

Tôi cũng xin tỏ lòng biết ơn về sự động viên giúp đỡ của gia đình và bạn bè, đồng nghiệp, nhừng người luôn quan tâm sát cánh bên tôi và là nguồn động viên khích lệ, tạo cho tôi có được những điều kiện tốt nhất trong suốt quá trình học tập của mình.

Tôi xin chân thành cảm ơn!

Hà Nội, ngày 10 tháng 6 năm 2022

**Học viên thực hiện**

(Ký tên và ghi rõ họ tên)

**Phạm Đức Tiến**

# **VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| PKI | Public Key Infrastructure | Hạ tầng khóa công khai |
| CA | Certification Authority | Tổ chức chứng thực |
| RA | Rigistration Authority | Tổ chức đăng ký |
| EE | End Entity | thực thể cuối |
| CRL | Certificate Revocation List | Danh sách hủy bỏ chứng nhận |
| DN | Distinguished Name | Tên phân biệt |
| PKCS | Public Key Cryptography Standard | Chuẩn mật mã khóa công khai |
| PDF | Portable Document Format | định dạng tập tin văn bản PDF |
| SHA | Secure Hash Algorithm | Thuật toán băm bảo mật |

# **MỞ ĐẦU**

Với sự phát triển mạnh mẽ của internet và hạ tầng CNTT ngày càng mở rộng, các tổ chức, người dùng sử dụng nền tảng CNTT để trao đổi và liên lạc với nhau. Các thông tin nhạy cảm, bí mật và quan trọng cũng được lưu trữ và trao đổi dưới hình thức điện tử để đảm bảo việc truyền nhận một cách nhanh chóng, tiện lợi. Sự phát triển này đã mở ra nhiều cơ hội cho các doanh nghiệp, tổ chức, nhưng cũng đem tới những thách thức không nhỏ, đặc biệt là sự can thiệp, tấn công phá hoại hoặc ý thức người dùng với các thông tin nhạy cảm và quan trọng. Trước vấn đề được đặt ra, cơ sở hạ tầng khóa công khai (PKI – Public Key Infrastructure) đang được coi là một giải pháp mang tính tổng hợp để giúp giải quyết các vấn đề bảo đảm an toàn dữ liệu nhạy cảm.

Trên thế giới PKI đang được hoàn thiện, đầu tư và xây dựng để dần hiện thực hóa, với nhiều chuẩn bảo mật trên mạng internet như SSL/TLS, VPN hay các thiết bị phần cứng HSM, USB Token thì PKI đang ứng dụng rộng rãi trong mọi hoạt động cuộc sống như giao dịch điện tử, chứng minh thư điện tử, hộ chiếu điện tử,… Và tầm quan trọng của chứng thư số và chữ ký số luôn là vấn đề mang tính thời sự trong mọi thời điểm.

Mục tiêu của việc nghiên cứu thực hiện đồ án tốt nghiệp với đề tài “*Xây dựng phần mềm ký số, mã mật file PDF trên Wed*” nhằm phát triển công cụ ký số, mã mật file PDF, có tính khả dụng cao, giúp cá nhân hay tổ chức đảm bảo tính chính xác, hợp pháp cũng như tính bí mật của tài liệu PDF.

Đồ án bao gồm 3 chương:

Chương 1 Hệ thống chứng thực số

Chương 2 Công nghệ ký số, mã mật file PDF và xây dựng ứng dụng web

Chương 3: Xây dựng ứng dụng Web ký số, mã mật file PDF

Sau thời gian khoảng ba tháng thực hiện đồ án tương đối ngắn nên chắc chắn đồ án của em không tránh khỏi thiếu sót. Em rất mong được sự góp ý của các thầy cô, cũng như các bạn học viên để đồ án này được hoàn thiện hơn.

Em xin chân thành cảm ơn!

|  |  |
| --- | --- |
|  | **SINH VIÊN THỰC HIỆN ĐỒ ÁN**  Phạm Đức Tiến |

# **HỆ THỐNG CHỨNG THỰC SỐ**

* 1. **Mật mã khóa công khai**
     1. ***Khái niệm***

Mật mã khóa công khai, còn được gọi là mật mã bất đối xứng, là một hệ thống sử dụng các cặp khóa để mã hóa và xác thực thông tin. Một khóa trong cặp là khóa công khai, được phân phối rộng rãi mà không ảnh hưởng đến bảo mật. Khóa thứ hai trong cặp khóa là khóa riêng, chỉ được chủ sở hữu biết.  
Việc sử dụng mật mã khóa công khai cung cấp cho ta những ứng dụng quan trọng trong việc bảo vệ thông tin:

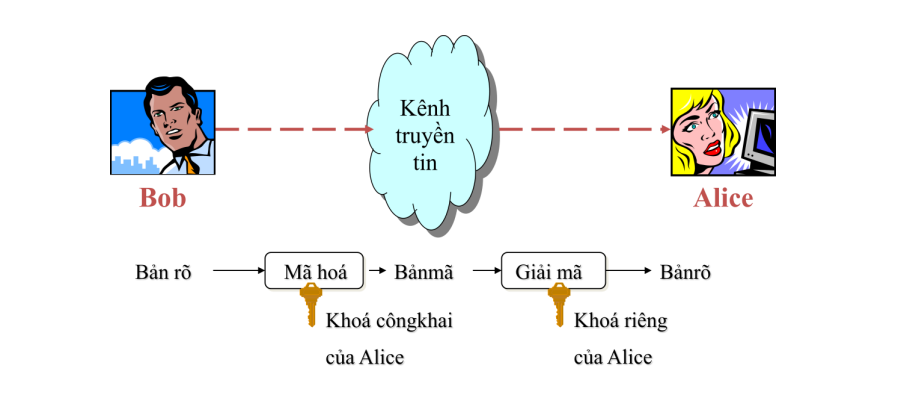
a. Bảo vệ tính bí mật của thông tin

Giả sử Bob muốn gửi cho Alice một thông điệp M, Bob sẽ phải thực hiện các bước sau:

- Mã hóa thông điệp M bằng khóa công khai của Alice

- Gửi bản mã thông điệp cho Alice

Khi Alice nhận được thông điệp đã được mã hóa của Bob, Alice sẽ dùng khóa riêng của mình để giải mã thông điệp đó.



Hình 1.1. Mật mã khóa công khai

Phương pháp này cung cấp tính bí mật vì chỉ có Alice mới có khóa bí mật để giải mã thành công bản mã mà Bob đã gửi. Tuy nhiên, phương pháp này lại không cung cấp bất kỳ quá trình xác thực nào để khảng định bản mã mà Alice nhận là do Bob gửi vì khóa công khai của Alice ai cũng biết.

b. Xác thực thông tin

Bob muốn mọi người biết tài liệu M là của chính Bob gửi, Bob có thể sử dụng khóa riêng của mình để ký lên tài liệu M.

Khi Alice nhận được tài liệu, Alice sẽ kiểm tra chữ ký có trong tài liệu M bằng khóa công khai của Bob và Alice biết chắc chắn được rằng tài liệu này là do Bob ký vì chỉ có Bob mới có khóa riêng dùng để ký lên tài liệu.

Phương pháp này giúp người sử dụng có thể xác thực được nguồn gốc của tài liệu, tuy nhiên tính bí mật của tài liệu không được bảo vệ. Do đó, người không được quyền xem tài liệu vẫn có thể xem được nó.

c. Bảo vệ tính bí mật và xác thực của thông tin

Để đảm bảo thông tin vừa bí mật, vừa xác thực, cần phải thực hiện mã hóa hai lần. - Trước tiên, Bob phải ký thông điệp bằng khóa riêng của mình (nhằm đảm bảo tính xác thực). - Sau đó, Bob sử dụng khóa công khai của Alice để mã hóa tiếp thông báo vừa được mã hóa (nhằm đảm bảo tính bí mật). Cuối cùng, Bob gửi bản mã đến Alice, Alice nhận được sẽ giải mã theo thứ tự ngược lại để lấy được bản rõ

* + 1. ***Các thuật toán sử dụng*** 
       1. Thuật toán RSA

Thuật toán RSA được Ron Rivest, Adi Shamir và Len Adleman mô tả lần đầu tiên vào năm 1977 tại Học viện Công nghệ Massachusetts (MIT). RSA là một thuật toán mã hóa khóa công khai. RSA là thuật toán khởi đầu của lĩnh vực mật mã trong việc sử dụng khóa công khai và phù hợp để tạo ra chữ ký điện tử.

RSA đang được sử dụng phổ biến trong thương mại điện tử và với độ dài khóa đủ lớn thì nó có thể được đảm bảo an toàn.

Độ an toàn của hệ thống RSA dựa trên 2 vấn đề của toán học: bài toán phân tích ra thừa số nguyên tố và bài toán RSA.

RSA có tốc độ thực hiện chậm hơn so với AES và các thuật toán mã hóa đối xứng khác. Trên thực tế, người ta sử dụng một thuật toán mã hóa đối xứng để mã hóa văn bản cần gửi còn RSA chỉ sử dụng để mã hóa khóa để giải mã

* + - 1. Phương thức trao đổi khóa Diffie-Hellman

Trao đổi khóa Diffie–Hellman (D-H) là một phương pháp trao đổi khóa được phát minh sớm nhất trong mật mã học. Phương pháp trao đổi khóa Diffie–Hellman cho phép người và thực thể giao tiếp thiết lập một khóa bí mật chung để mã hóa dữ liệu sử dụng trên kênh truyền thông không an toàn mà không cần có sự thỏa thuận trước về khóa bí mật giữa hai bên. Khóa bí mật tạo ra sẽ được sử dụng để mã hóa dữ liệu với phương pháp mã hóa khóa đối xứng.

Các thuật toán mã hóa khóa công khai cũng không được đảm bảo an toàn tốt hơn các thuật toán khóa bí mật. Vì vậy, cũng giống như tất cả các thuật toán mật mã nói chung, các thuật toán mã hóa khóa công khai cần phải được sử dụng một cách thận trọng.

* 1. **Chứng thực số** 
     1. ***Một số khái niệm***
        1. Cơ sở hạ tầng khóa công khai

Cơ sở hạ tầng khóa công khai (PKI) được phát triển dựa trên kỹ thuật mật mã khóa công khai nhằm đảm bảo các mục tiêu: bí mật, toàn vẹn, xác thực, chống chối bỏ. PKI là một hệ thống tập hợp bao gồm phần cứng, phần mềm, con người, chính sách và các thủ tục cần thiết để đào tạo, quản lý, phân phối, sử dung, lưu trữ, và thu

hồi các chứng thư số (chứng thư điện tử).  
Một số khái niệm khác:

PKI là cơ sở của một hạ tầng an ninh rộng khắp, các dịch vụ của PKI được cài đặt và thực hiện bằng cách sử dụng các khái niệm và kỹ thuật của mật mã khóa công khai. Trong mật mã khóa công khai, cơ sở hạ tầng khóa công khai là một cơ chế để cho bên thứ 3 cung cấp và xác thực định danh các bên tham gia vào quá trình trao đổi thông tin. Cơ chế này cũng cho phép gán cho mỗi người sử dụng trong hệ thống một cặp khóa công khai/khóa bí mật. Các quá trình này thường được thực hiện bởi một phần mềm đặt tại trung tâm và các phần mềm phối hợp khác tại các địa điểm của người dùng. Khóa công khai thường phân phối trong chứng thư khóa công khai.

Khái niệm hạ tầng khóa công khai thường được dùng để chỉ toàn bộ hệ thống bao gồm thẩm quyền chứng thực cùng các cơ chế liên quan đồng thời với toàn bộ việc sử dụng các thuật toán mã hóa công khai trong trao đổi thông tin.

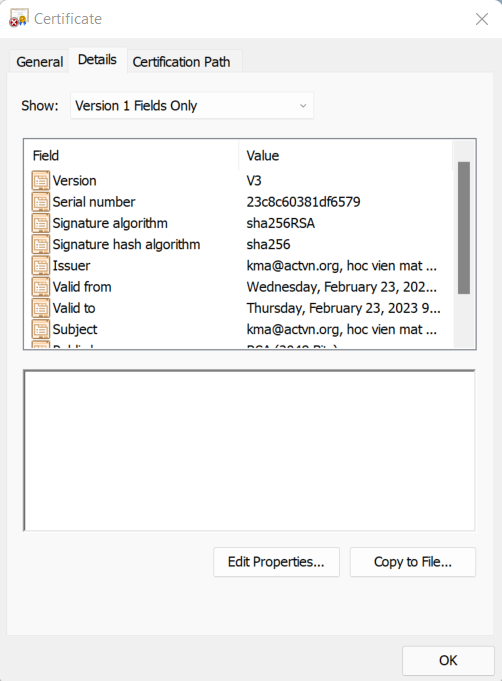
* + - 1. Chứng thực số

Chứng thực số là việc một tổ chức có thẩm quyền (tin cậy) sử dụng các công nghệ, kỹ thuật điện tử (số) chứng nhận một cặp khóa thuộc về một chủ thể thông qua phương tiện xác minh (chứng thư số)

Bản chất hệ thống chứng thực số là một hệ thống hỗ trợ cho việc áp dụng các kỹ thuật mật mã (đặc biệt mật mã khóa công khai) nhằm đảm bảo an toàn, tin cậy cho các giao dịch trong môi trường mạng

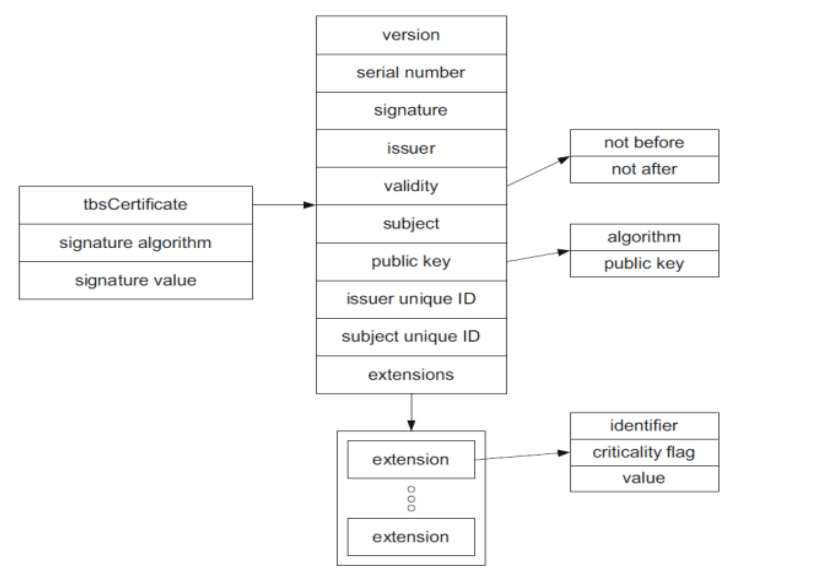
* + - 1. Chứng thư số

Chứng thư số là một dạng chứng thư điện tử, nó được cung cấp bởi tổ chức cung cấp dịch vụ chứng thực số. Chứng thư số được xem như là thẻ căn cước sử dụng trên môi trường mạng máy tính. Chứng thư số chứa tất cả các thông tin cần thiết như khóa công khai, chủ thể, bên cấp chứng thư số và không chứa bất kỳ một thông tin bí mật nào. Vì vậy, chứng thư số được dùng để nhận diện một cá nhân, một máy chủ và được gắn định danh đối tượng với khóa công khai. Bên cấp chứng thư số đảm bảo tính hợp lệ của chứng thư số thông qua chữ ký số. Nếu chữ ký đó hợp lệ thì người dùng có thể sử dụng chứng thư đó.



Hình 1.2. Ví dụ về chứng thư số

Hiện nay, chứng thư số X.509 là chứng thư số được sử dụng phổ biến trong hầu hết các hệ thống PKI. Chứng thư số X.509 có 3 phiên bản đó là:

* Chứng thư X.509 Version 1
* Chứng thư X.509 Version 2
* Chứng thư X.509 Version 3 

Hình 1.3. Cấu trúc chung chứng thư số X.509

Những trường cơ bản của chứng thư số X.509:

* Version: Phiên bản của chứng thư số. Ví dụ: V3
* Serial Number: Số serial của chứng thư số, là định danh duy nhất của chứng thư số, có giá trị nguyên.
* Signature Algorithm: Thuật toán của chữ ký: Chỉ ra thuật toán CA sử dụng để ký chứng thư số. Ví dụ: SHA-1 with RSA.
* Issuer: Tên phân biệt DN (Distinguished Name) của CA đã phát hành chứng thư số và phải luôn được hiện diện.
* Valid from – Valid to: Thời hạn của chứng thư số.
* Subject: Chỉ ra DN của chủ sở hữu chứng thư số và phài là non-null trừ khi một định dạng tên thay thế được sử dụng (đề cập tới trường Extensions sau).
* Subject Public Key Info: Là khóa công khai kết hợp với chủ thể và phải luôn luôn được hiện diện.
* Issuer Unique ID (Optional): Là một định danh duy nhất tùy chọn của người phát hành chứng thư số hiện diện chỉ trong phiên bản 2 và phiên bản 3. Trường này hiếm được sử dụng trong triển khai thực tế.
* Subject Unique ID (Optional): Là định danh duy nhất tùy chọn của chủ sở hữu chứng thư số hiện diện chỉ trong phiên bản 2 và phiên bản 3. Trường này hiếm được sử dụng trong triển khai thực tế.
* Extensions (Optional): Cung cấp những thông tin bổ sung, chỉ có trong phiên bản 3.
* Digital Signature: Bao gồm thuật toán mật mã (được chỉ ra trong trường Signature) được sử dụng để CA tạo ra chữ ký số, chữ ký số của CA thu được bằng việc băm tất cả những thông tin trên chứng thư và việc mã hóa nó với khóa riêng của CA.

Bản chất của chứng thư số là cấu trúc dữ liệu gắn với các thông tin xác định chủ thể với một khóa công khai, được ký số bởi cơ quan phát hành (Tổ chức chứng thực) và lưu trữ dưới dạng một tập tin (file).

Những phần mở rộng của tên tập tin phổ biến cho chứng nhận X.509 bao gồm:

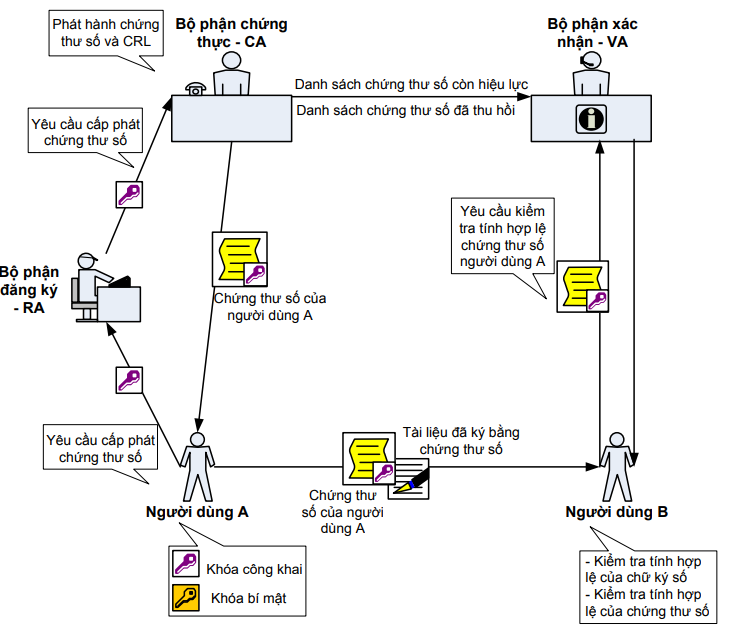
* cer: chứng nhận được mã hóa theo luật mã hóa tiêu chuẩn (Canonical Encoding Rules – CER).
* der: chứng nhận được mã hóa theo luật mã hóa phân biệt (Distinguished Encoding Rules – DER).
* pem (Privacy-Enhanced Electronic Mail): định dạng mã hóa được sử dụng để lưu trữ các chứng nhận và khóa. Một tập tin được định dạng với chuẩn này có thể chứa các khóa bí mật (RSA và DSA), khóa công khai (RSA và DSA) và các chứng nhận X509. Định dạng này lưu trữ dữ liệu ở định dạng DER được mã hóa cơ sở 64, nằm giữa "-----BEGIN CERTIFICATE-----" và "-----END CERTIFICATE-----", phù hợp cho việc trao đổi ở dạng văn bản giữa các hệ thống.
* p7b, p7c: PKCS #7 là một định dạng mã hóa cho việc lưu trữ một chứng nhận số và chuỗi chứng nhận của nó dưới dạng các ký tự ASCII. Định dạng này được sử dụng bởi CA để trả về các chứng nhận được phát hành cùng với chuỗi chứng nhận. Định dạng này có có thể được sử dụng như đầu vào cho yêu cầu gia hạn chứng nhận đến một CA.
* .pfx, .p12: PKCS #12 là một định dạng mã hóa cho việc lưu trữ một chứng nhận số và kết hợp với khóa bí mật dưới dạng các ký tự ASCII. Định dạng này luôn luôn được trả về bởi CA khi CA phát sinh các khóa và phát hành chứng nhận đồng thời

Phân loại:

Căn cứ vào đối tượng của chứng thư số, có thể phân loại thành 2 loại:

* Chứng thư số thực thể cuối: là loại chứng thư số do một CA phát hành cho thực thể cuối. Chứng thư này không dùng để phát hành chứng thư khác.
* Chứng thư số CA:là chứng thư số do một CA phát hành cho một CA được phân biệt bởi trường Basic Constraint. Chứng thư số CA này có thể dùng để phát hành chứng thư khác.
* Căn cứ vào mục đích sử dụng, chứng thư số có thể phân thành 3 loại:
* Chứng thư số khóa công khai ( Public Key Certificate)
* Chứng thư số thuộc tính
* Chứng thư số đủ điều kiện
* Card Verifiable Certificates( CVC)
  1. Các thành phần của hệ thống chứng thực số

Hệ thống chứng thực số là một hạ tầng an ninh mạng được xây dựng dựa trên một hạ tầng cơ sở khóa công khai ( PKI ).



Hình 1.4. Mô hình hoạt động của hệ thống CTS

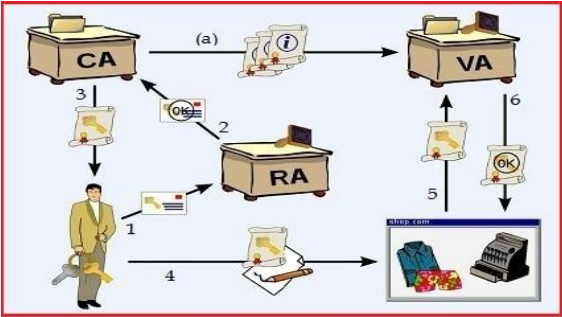
Một hệ thống chứng thực bao gồm :hạ tầng PKI và thực thể cuối. Trong đó:

Hạ tầng PKI:

* Certification Authority ( CA) : Cấp và thu hồi chứng thư số
* Registration Authority (RA) : gắn kết giữa khóa công khai và định danh của người giữ chứng thư số
* Validation Authority (VA) : Ủy quyền xác nhận hợp lệ, xác nhận tính hợp lệ thể chứng thư số của một đối tác trao đổi thông tin.
* Hạ tầng PKI cung cấp các dịch vụ:
* -Chứng thực: phát hành, thu hồi chứng thư số. Phát hành danh sách thu hồi (CRL). Quản lý vòng đời của chứng thư số sau khi phát hành.
* Các dịch vụ ( trực tuyến) nhằm khẳng định tính tin cậy và hợp lệ của chứng thư số : LDAP, CRL, OSCP,…
* Dịch vụ dấu thời gian: TimeStamp,..

Thực thể cuối:

* Chủ sỡ hữu chứng thư số (thuê bao).
* Người dùng, thiết bị, tin cậy và sử dụng chứng thư số



Hình 1.5. Các thành phần trong hệ thống PKI

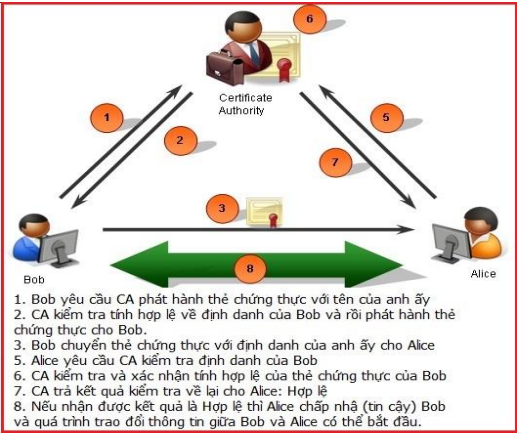
Người dùng gửi yêu cầu phát hành thẻ chứng thực và khóa công khai của nó đến RA (1); Sau khi xác nhận tính hợp lệ định danh của người dùng thì RA sẽ chuyển yêu cầu này đến CA (2); CA phát hành thẻ chứng thực cho người dùng (3); Sau đó người dùng “ký” thông điệp trao đổi với thẻ chứng thực mới vừa nhận được từ CA và sử dụng chúng (thẻ chứng thực số + chữ ký số) trong giao dịch (4); Định danh của người dùng được kiểm tra bởi đối tác thông qua sự hỗ trợ của VA (5): Nếu thẻ chứng thực của người dùng được xác nhận tính hợp lệ (6) thì ối tác mới tin cậy người dùng và có thể bắt đầu quá trình trao đổi thông tin với nó (VA nhận thông tin về các thẻ chứng thực đã được phát hành từ CA (a)).

* + 1. ***Thầm quyền chứng thực (CA- Certification Authority)***

Trong PKI, CA là một thực thể PKI có trách nhiệm cấp chứng thư số cho các thực thể khác trong hệ thống.

CA là thành phần thứ 3 tin cậy (trusted third part), nó nhận một yêu cầu phát hành (cấp) thẻ chứng thực, từ một tổ chức hoặc một cá nhân nào đó, và phát hành thẻ chứng thực yêu cầu đến họ sau khi đã xác thực client yêu cầu .

CA dựa vào các chính sách, trao đổi thông tin trong môi trường bảo mật, của tổ chức để định nghĩa một tập các quy tắc, các thủ tục liên quan đến việc phát hành thẻ chứng thực. Mọi họat động tạo, phát hành, thu hồi thẻ chứng thực sau này đều tuân theo các quy tắc, thủ tục này.



Hình 1.6. Quá trình xác thực dựa trên CA

CA có những nhiệm vụ chính:

* Gắn một cặp khóa công khai với một định danh đã cho.
* Chứng nhận việc gắn kết này bằng cách ký số một cấu trúc dữ liệu có chứa biểu diễn của định danh (gọi là chứng thư).
  + 1. ***Thẩm quyền đăng ký (RA- Registration Authority)***

Chức năng quản trị có thể được phân phối cho RA. RA đóng vai trò trung gian làm nhiệm vụ tương tác giữa CA và client .

RA có thể chịu trách nhiệm cho việc gán tên, tạo cặp khóa, xác thực thực thể cuối trong suốt quá trình đăng ký.

RA làm nhiệm vụ nhận các yêu cầu của thực thể, xác minh chúng sau đó gửi yêu cầu cho CA và RA cũng nhận các chứng chỉ từ CA, sau đó gửi chứng thư cho thực thể. Thiết lập và xác nhận danh tính của thực thể trong giai đoạn khởi tạo phân phối các bí mật dùng chung tới người sử dụng cuối để xác thực tuần tự trong suốt giai đoạn khởi tạo trực tuyến.

Khởi tạo quá trình chứng thực với CA đại diện cho người dùng cuối

Thực hiện chức năng quản lý vòng đời của khóa/chứng chỉ, tuy nhiên, CA không bao giờ được phép cấp chứng thư số hoặc thu hồi chứng thư số. Chức năng này chỉ có ở RA.

Thành phần của RA bao gồm 3 thành phần như sau :

- RA console là một máy chủ được cài đặt cho RA officer để đưa các yêu cầu chứng chỉ. Nó có thể kết nối với CA. Máy chủ này xử lý các yêu cầu chứng thư số trong quá trình chứng thực.

- RA Officer là một cá nhân thực hiện các tác vụ như đăng ký chứng thư số, làm mới hoặc thu hồi chứng thư số. Sau khi RA Officer xác minh và chấp thuận yêu cầu, nó sẽ chuyển trực tiếp các yêu cầu này lên CA server. Sau khi CA server xử lý yêu cầu và cấp chứng thư số. RA Officer sẽ phân phối chứng thư số.

- RA Manager là một cá nhân làm nhiệm vụ quản lý RA Officer và đảm bảo rằng toàn bộ thủ tục ứng dụng chứng thực được thực hiện mà không có sự lừa đảo của con người. RA Manager sẽ cần phải chấp thuận tất cả các yêu cầu được xử lý bởi RA Officer trước khi đưa các ứng dụng chứng thực tới cho CA.

Mục đích chính của RA là để giảm tải công việc của CA. Chức năng thực hiện của một RA cụ thể sẽ khác nhau tùy theo nhu cầu triển khai PKI nhưng chủ yếu bao gồm các chức năng sau:

- Xác thực cá nhân chủ thể đăng ký chứng thư.

- Kiểm tra tính hợp lệ của thông tin do chủ thể cung cấp.

- Xác nhận quyền của chủ thể đối với những thuộc tính chứng thư được yêu cầu.

- Kiểm tra xem chủ thể có thực sự sở hữu khoá riêng đang được đăng ký hay không - điều này thường được đề cập đến nhờ sự chứng minh sở hữu (proof of possession - POP).

- Tạo cặp khoá bí mật /công khai.

- Phân phối bí mật được chia sẻ đến thực thể cuối (ví dụ: khoá công của CA).

- Thay mặt chủ thể thực thể cuối khởi tạo quá trình đăng ký với CA.

- Lưu trữ khoá riêng.

- Khởi sinh qúa trình khôi phục khoá.

- Phân phối thẻ bài vật lý (ví dụ như thẻ thông minh) chứa khoá riêng.

* + 1. ***Thẩm quyền xác thực (VA- Validation Authority)***

Thẩm quyền xác thực thay mặt CA xác nhận thông tin nhận dạng của người dùng và khẳng định tính tin cậy, hợp lệ của các cặp khóa đã chứng thực thuộc/không thuộc đối với các giao dịch.

VA đóng vai trò là: kho công cộng chứa chứng thư số: CA phát hành chứng thư, gửi cho người dùng và lưu trữ nó vào kho chứng thư (Certificate Repository). Kho chứng thư được CA công bố và được truy cập bằng nhiều giao thức khác nhau (http,ftp...)

Chức năng của VA là công bố gốc tin cậy của hệ thống (công bố các chứng thư số gốc, các chứng thư chéo, danh sách tin cậy), công bố thông tin chứng thực cặp khóa (công bố chứng thư số CA trung gian, chứng thư số người dùng cuối, công bố CARL và CRL) và cung cấp các dịch vụ hỗ trợ đảm bảo an toàn cho các giao dịch điện tử.

Các thành phần của RA:

* Kho chứng thư số: chứa tất cả các chứng thư số, CARL, CRL do tổ chức chứng thực tạo ra, điểm tin cậy của hệ thống.
* Các dịch vụ hỗ trợ kiểm tra trạng thái chứng thư số: OSCP, SCVP
* Dịch vụ dấu thời gian an toàn: tạo dấu thời gian gắn kèm với dữ liệu thỏa mãn các tính chất xác thực và toàn vẹn.
  + 1. ***Thực thể cuối (EE- End Entity)***

Thực thể cuối là những đối tượng sử dụng cặp khóa đã chứng thực để đảm bảo an toàn cho các giao dịch điện tử.

Thực thể cuối không chỉ là người sử dụng mà còn bao gồm những thứ vô tri vô giác như máy tính, những đối tượng cần chứng nhận số để nhận biết chúng vì một số lý do nào đó. Thực thể cuối thông thường phải có khả năng phát sinh cặp khóa công khai/ bí mật và một số phương tiện cho việc lưu trữ và sử dụng khóa bí mật một cách an toàn.

Thực thể cuối gồm 2 loại:

* Chủ thể tin cậy và sử dụng chứng thư số - relying party
* Chủ thể sở hữu chứng thư số - subcriber

Thực thể cuối giao tiếp với hạ tầng (các dịch vụ mà hạ tầng cung cấp) thông qua phần mềm người dùng cuối (client-side software) còn gọi là PKI-Client software.

Các thực thể yêu cầu CA hoặc RA để yêu cầu chứng thư là PKI client. Để đạt được chứng thư từ CA, PKI Client thường phải thực hiện các bước sau:

* Gửi một yêu cầu sinh cặp khóa bí mật – công khai
* Sau khi cặp khóa được sinh ra, một yêu cầu được gửi tới CA để sinh chứng thư. Yêu cầu này có thể thông qua RA.
* Sau khi client nhận chứng thư từ CA, họ có thể sử dụng chứng thư để định danh cho chính bản thân hoặc dùng để xác thực người nắm giữ chứng thư.
  + 1. ***Kho chứng thư số***

Khi A và B thực hiện các giao dịch điện tử được xác thực và bảo mật bởi hạ tầng khóa công khai PKI, A và B đều phải lấy chứng thư số liên kết với khóa công khai của nhau để thực hiện các giao dịch. Đối với mô hình mạng giao dịch nhỏ, ít người dùng chứng thư số của người dùng có thể được công bố và phân phối tùy ý người sử dụng, trước khi giao dịch thì người dùng phải thực hiện tìm kiếm chứng thư số của đối tác để thực hiện các giao dịch an toàn. Đối với mô hình mạng giao dịch lớn, nhiều người dùng phân tán, việc công bố và phân phối chứng thư số phải được thực hiện tập trung có sự quản lý chặt chẽ qua Kho chứng thư số.

Các dạng Kho chứng thư số thường sử dụng : X.500, LDAP, Active Directory, Web, Cơ sở dữ liệu, các máy chủ FTP,

* + 1. ***Hủy bỏ chứng thư số***

Thẩm quyền chứng thực CA ký lên một chứng thư để gắn một cặp khoá công khai với định danh của người sử dụng. Trong quá trình sử dụng chứng thư số có thể người sử dụng làm mất khóa bí mật, thay đổi một số thuộc tính định danh, khóa bí mật bị đánh cắp,… Cần phải có một cách để cảnh báo cho những người sử dụng khác trong hệ thống để họ không tiếp tục sử dụng chứng thư số đó để giao dịch. Cơ chế cảnh báo này trong một PKI được gọi là huỷ bỏ chứng thư.

* + 1. ***Sao lưu và khôi phục khóa***

Trong một số tường hợp, thiết bị lưu khóa của người dùng bị hỏng, khóa cài trên máy tính và máy tính gặp sự cố, khóa bị xóa, người dùng không thể sử dụng khóa để giải mã các dữ liệu đã mã trước đó. Một số trường hợp người dùng sử dụng khóa mã mã hóa các tài liệu vi phạm đến an ninh quốc gia, các cơ quan an ninh yêu cầu CA phải giải mã tài liệu. Một số tài liệu mã được lưu trữ lâu dài, khi đó chứng thư số đã hết hạn sử dụng, người dùng muốn lấy lại các khóa cũ để giải mã tài liệu trữ trữ….

Giải pháp sao lưu và phục hồi khóa là rất cần thiết, giải pháp này chỉ phù hợp với khóa mã tương ứng với chứng thư số sử dụng để mã dữ liệu, các loại chứng thư số khác không cần sao lưu (chứng thư số ký, xác thực).

* + 1. ***Cập nhật khóa tự động***

Một chứng thư có thời gian sống hữu hạn. Khi chứng thư số hết hạn, để tiếp tục sử dụng người dùng phải yêu cầu cấp một chứng thư số khác.

Việc cập nhật khóa tự động là khá thuận tiện cho người sử dụng, tuy nhiên nó sẽ chỉ phù hợp với một số chứng thư số như chứng thư số ký, xác thực, đối với chứng thư số mã, khi thay khóa mới sẽ có nhiều vấn đề phát sinh với các dữ liệu mã với khóa cũ, với chứng thư số mã đã công khai… Hơn nữa việc cập nhật khóa tự động cũng có thể là kẽ hở an ninh để tội phạm mạng có thể khai thác chặn bắt khóa trong quá trình cập nhật.

* + 1. ***Lịch sử khóa***

Một người dùng trong quá trình sử dụng chứng thư số có thể cập nhật khóa nhiều lần (do chứng thư số hết hạn, hủy bỏ,…), việc cập nhật có thể thủ công hoặc tự dộng, kéo theo rằng, trên toàn bộ diễn biến thời gian, một người sử dụng đã cho có nhiều chứng thư “cũ” và ít nhất một chứng thư “hiện tại đang sử dụng”. Tập hợp các chứng thư này và các khoá bí mật tương ứng được gọi là “lịch sử khoá” (key history) của người dùng (một cách đúng hơn là lịch sử chứng thư và khoá).

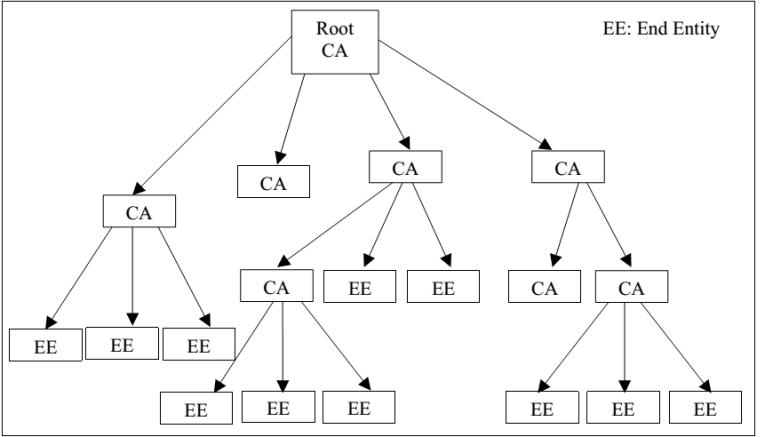
Giống như việc cập nhật khoá, việc quản trị các lịch sử khoá cũng cần phải tự động và hoàn toàn được duy trì bởi PKI. Những người dùng thông thường sẽ không cần duy trì bất kỳ một hệ thống lưu trữ khóa nào để họ có thể chọn khoá bí mật cần thiết, họ cũng có thể thử mỗi khoá bí mật lần lượt cho đến khi được giải mã được dữ liệu. PKI cần nắm giữ tất cả các khoá trong lịch sử, thực hiện sao lưu và khôi phục ở những nơi thích hợp, và tìm được khoá thích hợp tương ứng với bất kỳ dữ liệu nào đã được bảo vệ.

* 1. **Các mô hình kiến trúc tin cậy**

Hiện nay PKI được triển khai trong nhiều tổ chức như là một công cụ đảm bảo những nguồn tài nguyên nhạy cảm an toàn. Tuy nhiên, với nhiều mục đích khác nhau, tiến trình khác nhau nên khó có thể đưa ra một tiêu chuẩn thiết kế chung. Về cơ bản có các mô hình kiến trúc PKI có dựa trên các mô hình chính: mô hình phân cấp, mô hình mạng lưới, mô hình danh sách tin cậy,...

* + 1. ***Mô hình phân cấp***

Mô hình phân cấp có dạng hình cây. Có RootCA ( CA gốc) ở mức cao nhất và là gốc tin cậy duy nhất của toàn bộ thực thể bên dưới. Dưới RootCA là các nhánh được mở rộng xuống dưới và là thực thể hoặc một số CA trung gian tạo các đỉnh trong của cây. Các lá của cây là thực thể (thường là end entity).



Hình 1.7. Mô hình phân cấp

Trong mô hình này RootCA cung cấp chứng thư cho các CA hoặc thực thể ngay dưới nó. Các CA này lại cung cấp chứng thư cho các thực thể hoặc nhiều CA khác ngay dưới nó. Tất cả các đối tượng đều phải biết khóa công khai của RootCA và tất cả các chứng thư đều có thể kiểm tra bằng cách kiểm tra đường dẫn của chứng thư đó tới RootCA.

Đặc biệt có thể thêm CA mới vào hệ thống.

*Ưu điểm:*

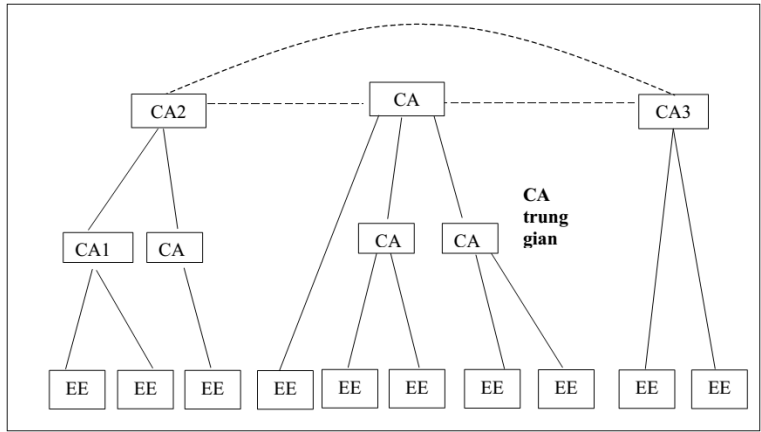
Mô hình này tương thích với cấu trúc phân cấp của hệ thống quản lý trong các tổ chức. Dễ làm quen do gần giống với hình thức phân cấp trong tổ chức thư mục. Mô hình không xảy ra hiện tượng vòng lặp do cách tìm nhánh theo một hướng nhất định nên đơn giản và nhanh.

*Nhược điểm:*

Trong một phạm vi rộng, một CA duy nhất không thể đảm nhận được tất cả quá trình xác thực. Do chỉ có CA gốc điều khiển toàn bộ kiến trúc PKI phân cấp nên nếu khóa riêng của CA gốc bị phá vỡ thì toàn bộ hệ thống PKI phân cấp sẽ bị nguy hiểm. Đối với các quan hệ kinh doanh thương mại thường không phải bao giờ có dạng phân cấp.

* + 1. ***Mô hình mạng lưới***

Trong mô hình này các CA xác thực ngang hàng tạo nên một mạng lưới tin cậy lẫn nhau. Các CA kề nhau cấp chứng chỉ cho nhau và CA này có thể xác thực CA kia theo nhiều nhánh khác nhau.



Hình 1.8. Mô hình mạng lưới

*Ưu điểm:*

Đây là mô hình linh động, thích hợp với các mối liên hệ tin cậy lẫn nhau trong thực tế và công việc kinh doanh. Mô hình này cho phép các CA xác thực ngang hàng trực tiếp, điều này đặc biệt có lợi khi các đối tượng sử dụng của các CA làm việc với nhau thường xuyên dẫn đến giảm tải lượng đường truyền và 46 thao tác xử lý. Khi một CA bị lộ khóa chỉ cần cấp phát chứng thư số của CA tới các đối tượng có thiết lập quan hệ tin cậy với CA này.

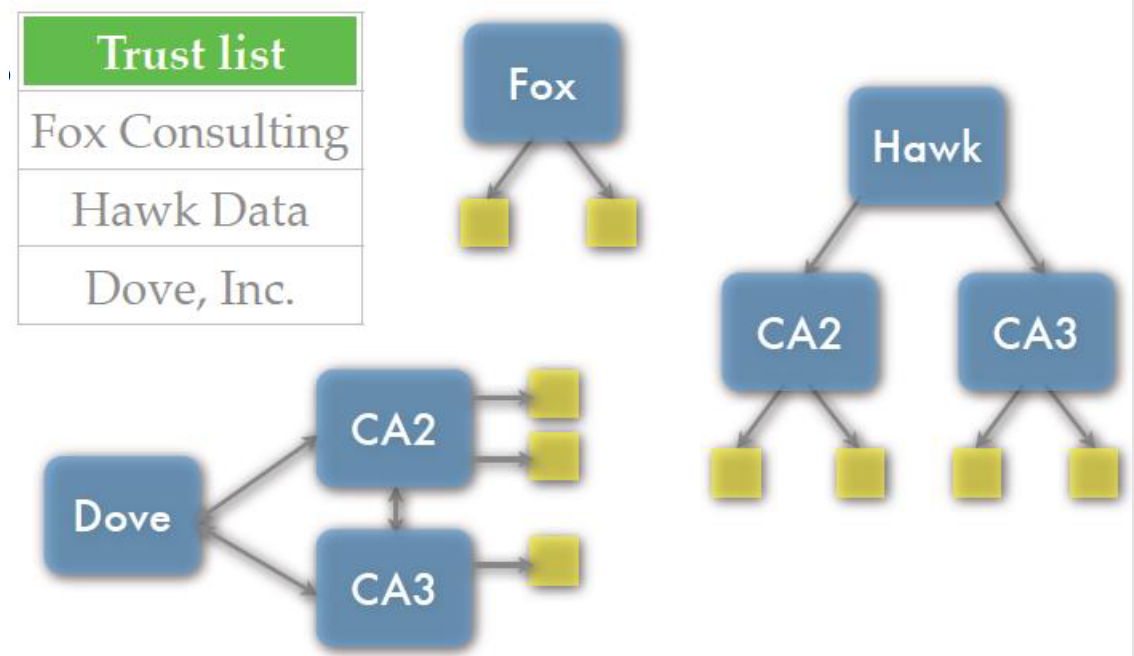
*Nhược điểm:*

Do mô hình mạng lưới có cấu trúc của mạng tương đối phức tạp nên việc tìm kiếm các đối tượng có thể khó khăn. Một đối tượng không thể đưa ra một nhánh xác thực duy nhất có thể đảm bảo rằng tất cả các đối tượng trong hệ thống có thể tin cậy được.

* + 1. ***Mô hình danh sách tin cậy***

Mô hình danh sách tin cậy là biến thể của mô hình phân cấp nhiều gốc- mô hình trust list.

Mô hình trust list bao gồm nhiều miển CA phân cấp độc lập. Các CA gốc được nhóm vào một danh sách tin cậy ( trust list) tạo thành một miền CA mới.



Hình 1.9. Mô hình danh sách tin cậy

*Ưu điểm:*

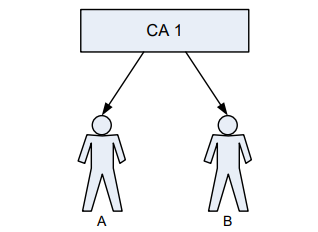
Mô hình danh sách tin cậy có kiến trúc đơn giản, dễ triển khai. Trong danh sách các CA là đối tượng tin cậy thì các đối tượng sử dụng hoàn toàn tin tưởng vào danh sách này. Các CA được tin cậy các đối tượng làm việc trực tiếp với CA.

*Nhược điểm:*

Mô hình danh sách tin cậy khó khăn trong việc quản lý danh sách các CA tin cậy. Việc tìm ra các nhánh xác nhận không được hỗ trợ nhiều từ cấu trúc chứng thư số.Cặp chứng thư số ngang hàng không được hỗ trợ trực tiếp vì vậy tạo ra hạn chế của CA trong việc quản lý sự tin cậy của mình với các CA khác.

* + 1. ***Mô hình CA đơn***

Mô hình kiến trúc CA đơn là mô hình cơ bản nhất của PKI. Mô hình này chỉ có duy nhất một CA được quyền cấp phát và phân phối chứng thư và danh sách thu hồi chứng thư tới các thực thể. Tất cả các thực thể hoàn toàn tin tưởng CA này. Hơn nữa các thực thể chỉ sử dụng chứng thư số do CA này cấp phát. Do chỉ có một CA duy nhất trong mô hình nên không tồn tại mối quan hệ với các CA tin cậy khác, và không cho phép bất kỳ 1 CA mới nào được phép thêm vào trong hệ thống PKI



Hình 1.10. Mô hình CA đơn

*Ưu điểm:*

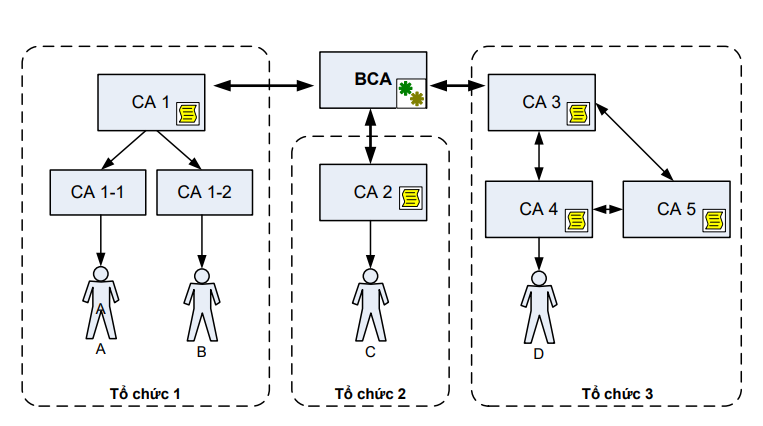
Mô hình này dễ để triển khai và giảm tối thiểu được những vấn đề về khả năng tương tác.

*Nhược điểm:*

Mô hình này không thích hợp cho miền PKI lớn vì một số người sử dụng ở những miền con có những yêu cầu khác nhau đối với người ở miền khác. Nhiều tổ chức không muốn vận hành hoặc không tin tưởng vào người vận hành cho CA đơn này vì việc quản trị và khối lượng công việc kỹ thuật của việc vận hành CA đơn sẽ rất cao trong cộng đồng PKI lớn. Nếu chỉ có một CA sẽ gây ra thiếu khả năng hoạt động và CA này có thể trở thành mục tiêu tấn công

* + 1. ***Mô hình cầu nối***

Kiến trúc CA cầu nối là kiến trúc phù hợp nhất để kết nối các PKI có kiến trúc khác nhau. Không giống như kiến trúc CA mạng lưới, kiến trúc CA cầu nối không cấp phát chứng thư số trực tiếp cho người dùng. CA cầu nối thiết lập mối quan hệ tin cậy ngang hàng với các CA của các thực thể khác nhau



Hình 1.11. Mô hình cầu nối

*Ưu điểm:*

Mô hình này làm giảm số xác thực chéo từ n^2 xuống n.Trong mô hình này, thay vì thiết lập xác thực chéo giữa các CA, mỗi CA gốc thiết lập xác thực chéo với CA trung tâm. CA trung tâm này làm cho việc giao tiếp được thuận lợi hơn. CA trung tâm được gọi là hub (hoặc bridge) CA . Với cấu hình này CA trung tâm không tạo ra sự phân cấp. Tất cả các thực thể trong cấu hình đều giữ khoá công khai của CA cục bộ, không có khoá của CA trung tâm.

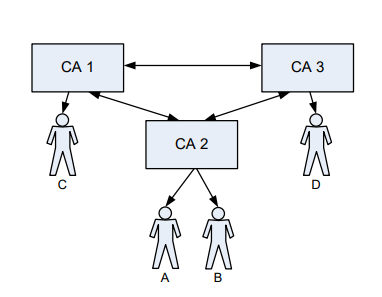
*Nhược điểm:*

Khó khăn trong việc thiết lập bridge CA làm việc với các CA khác trong hạ tầng cơ sở để các CA này có thể hoạt động được với nhau.

* + 1. ***Chứng thực chéo***

Chứng thực chéo là thiết lập mối quan hệ tin cậy ngang hàng giữa 2 CA.

CA gốc hoặc CA cấp con của một PKI thiết lập quan hệ tin cậy ngang hàng với các CA gốc hoặc các CA cấp con của PKI khác, tức là các CA gốc của mỗi nhóm sẽ cấp chứng nhận cho nhau.



Hình 1.12. Mô hình chứng thực chéo

Mỗi người dùng có một điểm tin cậy đơn, quan hệ chứng thực chéo là quan hệ ngang hàng. Khi các CA chứng thực chéo với nhau, các thực thể có thể xác thực các thực thể ở hệ thống PKI khác. Vì thế, có thể liên lạc an toàn giữa tất cả các thực thể qua các PKI.

* 1. **Các hoạt động chính của hệ thống chứng thực số** 
     1. ***Chứng thực cặp khóa***

Cấp mới, gia hạn, thay đổi nội dung thông tin, thu hồi chứng thư số, khôi phục thiết bị lưu khóa bí mật.

* Chứng thư số cho cá nhân.
* Chứng thư số cho cơ quan, tổ chức.
* Chứng thư số cho thiết bị, dịch vụ, phần mềm.



Hình 1.13. Sơ đồ chứng thực cặp khóa

* + 1. ***Khẳng định tính hợp lệ, tin cậy***

Kiểm tra tình trạng của chứng thư số dựa trên CRL hoặc OCSP

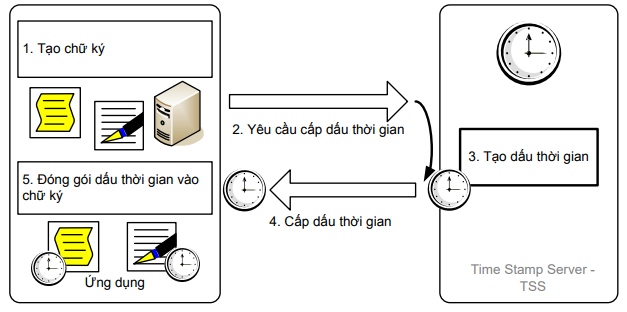
Việc xác minh tính hợp lệ của chứng thư số gồm các bước sau:

* Kiểm tra được rằng chứng thư số còn hiệu lực
* Xác minh được chứng thư số đó do một CA tin tưởng cấp phát
* Đảm bảo rằng chứng thư số đó được sử dụng vào đúng mục đích
* Kiểm tra, xác định được chứng thư số có bị thu hồi hay không.
* Một kịch bản sau đây mô tả quá trình xác minh tính hợp lệ của chứng thư số:
* D cần xác minh tính hợp lệ chứng thư số của G. Bước đầu tiên, D thực hiện các công việc
* Kiểm tra thời gian sống của chứng thư số để đảm bảo chắc chắn rằng chứng thư số không bị hết hạn
* Kiểm tra thông tin chính sách trong chứng thư số để chắc chắn chứng thư số được cấp phát cho G là không bị sử dụng sai mục đích Tiếp theo, D kiểm tra tính xác thực của chứng thư số:
* Kiểm tra danh sách CA tin cậy của D để tìm xem CA của G có trong danh sách đó không. Nếu CA của G không có trong danh sách, D tiếp tục tìm kiếm CA đó trong đường dẫn chứng thực trong CA phân cấp
* Sau khi tìm được CA tin cậy đó, D kiểm tra chữ ký số trong chứng thư số của G. Để xác minh chữ ký số, D tạo ra một hàm băm cho thông điệp đó. Sau đó, sử dụng khóa công khai để giải mã giá trị băm mà G gửi. Cuối cùng, D so sánh giá trị băm được giải mã với giá trị băm mà anh ta tạo ra. D lấy chứng thư khóa công khai của CA cấp phát cho G, sử dụng thuật toán băm trên thông điệp để tính toán giá trị băm và so sánh với giá trị hăm vừa được giải mã. Nếu hai giá trị trùng nhau, thì chứng thư số không bị sửa đổi và xác thực. Sau khi D xác thực được chứng thư số của G, D sẽ lấy danh sách thu hồi chứng thư CRL mới nhất để xác minh rằng chứng thư số này chưa bị thu hồi vì bất kỳ lý do nào
  + 1. ***Dịch vụ cấp dấu thời gian***

Dấu thời gian (time stamping) là một đối tượng dữ liệu gắn kết một dữ kiện với một mốc thời gian xác định, giúp biểu thị rằng một tài liệu nào đó là trước tài liệu X và sau tài liệu Y. Hệ thống phải có khả năng kiểm tra rằng dấu thời gian được liên kết với dữ liệu là xác thực và có tính toàn vẹn. Dấu thời gian bao gồm chữ ký số trên tổ hợp của:

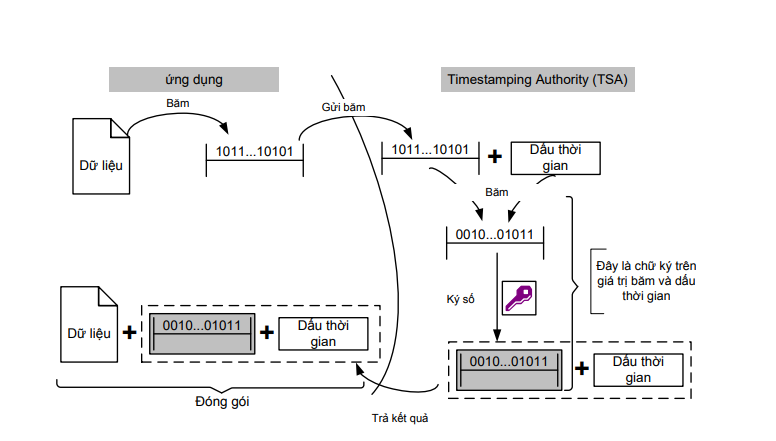
* Dữ liệu biểu diễn thời gian
* Giá trị băm của chính tài liệu.

Các thực thể PKI liên quan cần biết và tin cậy bằng chứng khóa công khai của dịch vụ dấu thời gian, bằng cách ký trên dấu thời gian có thể được kiểm tra và tin cậy; tất cả các dấu thời gian được ký bằng khóa không tin cậy được xem là không hợp lệ.



Hình 1.14. Sơ đồ cấp dấu thời gian

Dịch vụ dấu thời gian an toàn có thể sử dụng các dịch vụ cốt lõi của PKI. Dấu thời gian an toàn là dấu thời gian thỏa mãn các tính chất xác thực và toàn vẹn. Dịch vụ cấp dấu thời gian sẽ cấp dấu thời gian cho các thực thể cuối được thực hiện bởi một bên tin cậy thứ ba gọi là thẩm quyền dấu thời gian (TSA), thẩm quyền dấu thời gian không yêu cầu chỉ có duy nhất mà có thể mỗi một môi trường cục bộ có một thẩm quyền dấu thời gian để phục vụ dịch vụ này. Dấu thời gian được mã hóa dưới dạng ASN.1 và được truyền qua các phương tiện như file, email, TCP, HTTP. Dịch vụ dấu thời gian an toàn sử dụng các dịch vụ cốt lõi của PKI về xác thực và toàn vẹn.



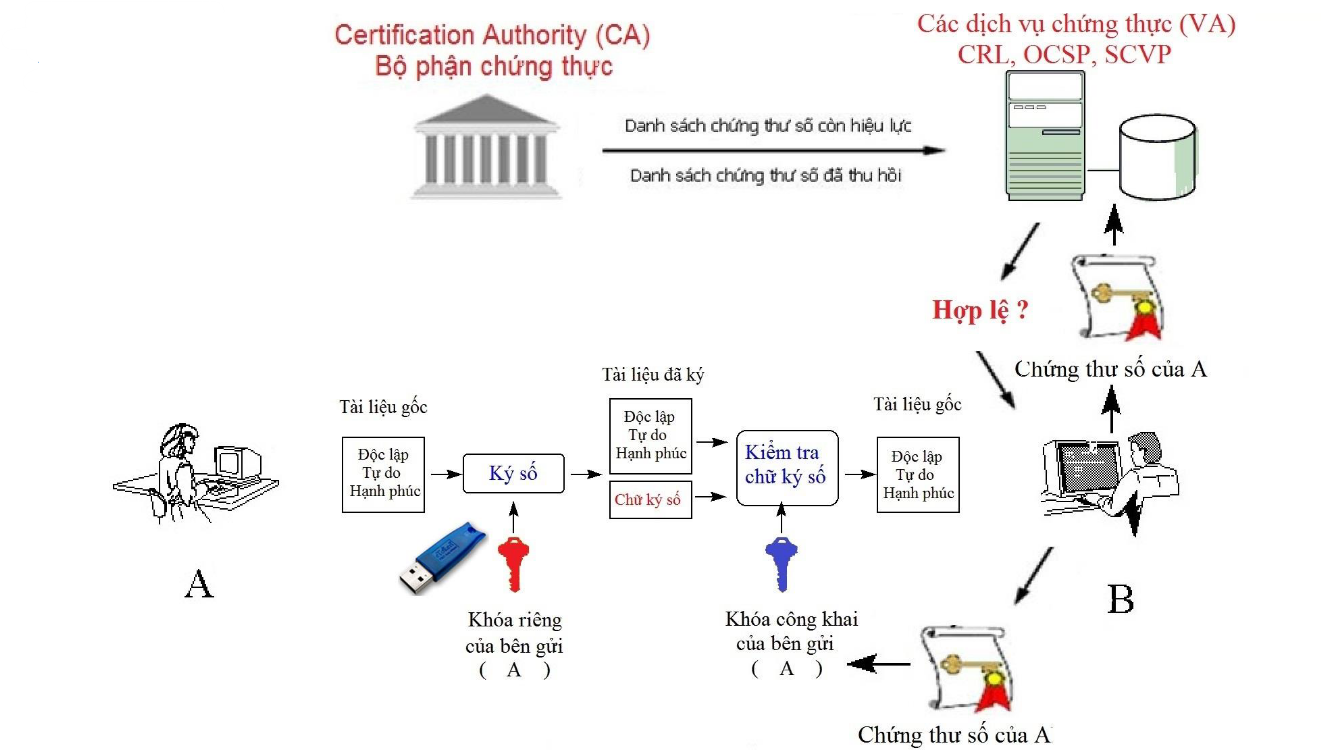
Hình 1.15. Mô hình tạo dấu thời gian an toàn



Hình 1.16. Kiểm tra tính hợp lệ của dấu thời gian an toàn

* 1. **Ứng dụng ký số**

Chữ ký số là một kết quả quan trọng của mật mã khóa công khai, về định nghĩa chữ ký số là một tập con của chữ ký điện tử. Chữ ký điện tử được tạo lập đưới dạng từ, chữ, số, ký hiệu, âm thanh hoặc các hình thức khác bằng phương tiện điện tử, gắn liền hoặc kết hợp một cách logic với thông điệp dữ liệu, có khả năng xác nhận người ký thông điệp dữ liệu và xác nhận sự chấp thuận của người đó đối với nội dung thông điệp dữ liệu được ký. Còn "chữ ký số" là một dạng chữ ký điện tử được tạo ra bằng sự biến đổi một thông điệp dữ liệu sử dụng hệ thống mật mã khóa công khai theo đó người có được thông điệp dữ liệu ban đầu và khoá công khai của người ký có thể xác định được chính xác



Hình 1.17. Ứng dụng ký số

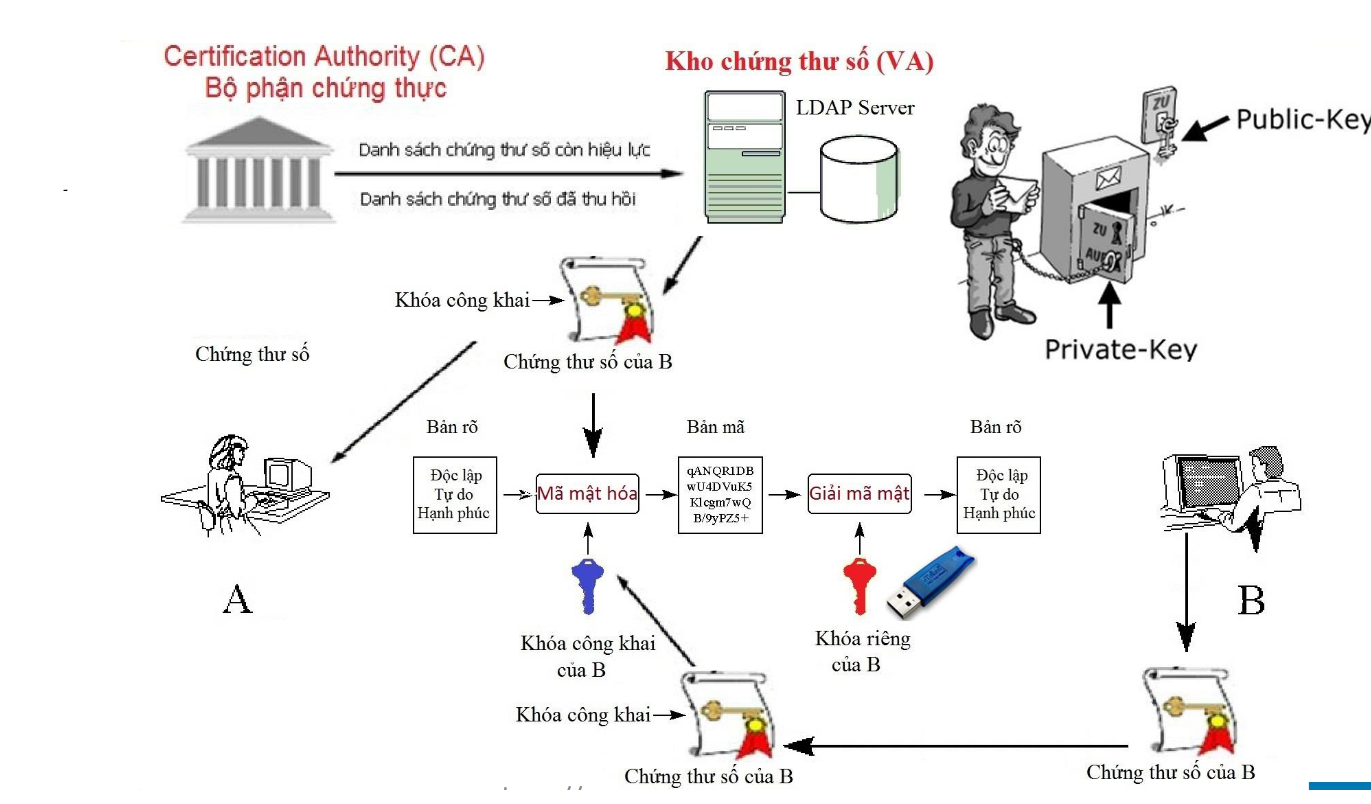
Chữ ký số vừa phục vụ mục đích cung cấp tính xác thực (authenticity) (tức là, xác thực thực thể), vừa cung cấp tính toàn vẹn trên dữ liệu được ký.

* 1. **Ứng dụng mã mật**

Ứng dụng mã mật là sử dụng tính đảm bảo bí mật của hệ thống PKI.

Dịch vụ đảm bảo tính bí mật là đảm bảo tính bí mật của dữ liệu: dữ liệu chỉ có thể truy nhập bởi những người có quyền, không ai đọc được nội dung của dữ liệu ngoại trừ những người dùng có quyền được đọc.

Tính bí mật này được cung cấp bởi các cơ chế mã hóa mật mã, bằng cách sử dụng cả mật mã khóa công khai và mật mã khóa bí mật. Do tốc độ mã hóa và giải mã thấp khi kích thước bản mã lớn nên mật mã khóa công khai không hiệu quả bằng mật mã khóa bí mật trong việc mã hóa dữ liệu lớn, nó thường được sử dụng để mã hóa những đối tượng dữ liệu nhỏ như các khóa bí mật được sử dụng trong các hệ thống mã hóa bất đối xứng.



Hình 1.18. Ứng dụng mã mật

Nhiệm vụ:

- Đảm bảo tính bí mật của dữ liệu chống lại các tấn công đọc trộm, nghe lén dữ liệu bởi những người dùng bất hợp pháp

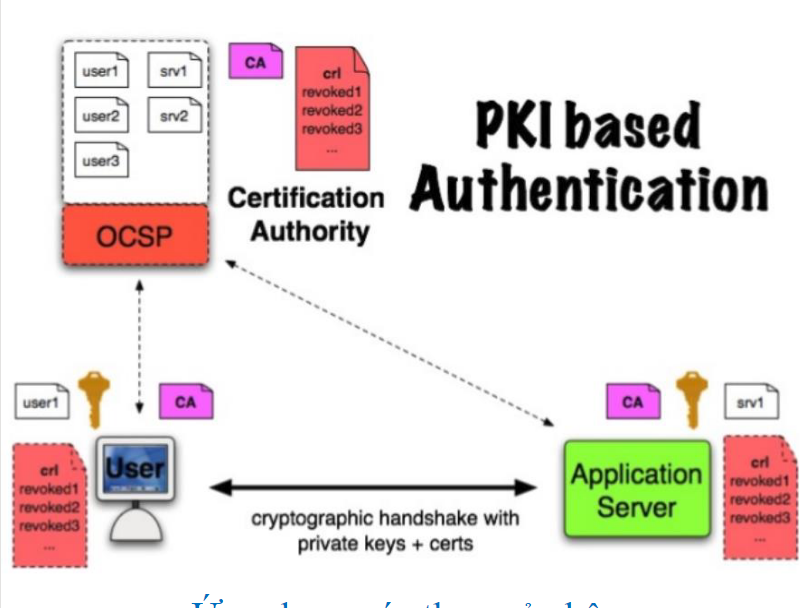
- Các dữ liệu nhạy cảm hoặc những dữ liệu được quy định là mật trở lên cần được bảo mật.

- Để đảm bảo tính bí mật, các thuật toán thích hợp và khóa sẽ được thỏa thuận giữa các bên để có thể mã hóa và giải mã chính xác dữ liệu

* 1. **Ứng dụng xác thực chống chối bỏ**

Xác thực nghĩa là danh tính của thực thể được xác minh. Tính xác thực trong môi trường thương mại điện tử được thực hiện rất tốt bằng các hệ thống mã hóa khóa công khai, dựa trên mối quan hệ toán học giữa khóa công khai và khóa bí mật. Thông điệp được ký bởi một thực thể có thể được kiểm tra bởi bất kỳ thực thể nào quan tâm.

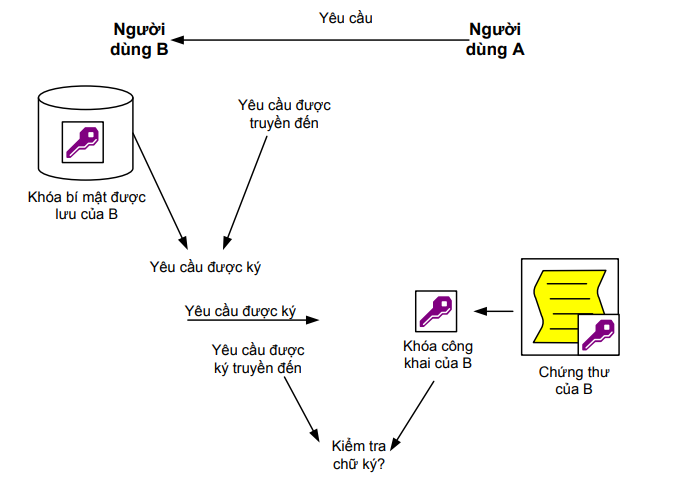
Dịch vụ đảm bảo tính xác thực là hành động hay khẳng định điều gì đó đáng tin cậy, là một thủ tục của một thực thể xác minh các tính chất đã khẳng định về một thực thể khác. Tính xác thực trong môi trường thương mại điện tử được thực hiện rất tốt bằng các hệ thống mã hóa khóa công khai, dựa trên mối quan hệ toán học giữa khóa công khai và khóa bí mật. Thông điệp được ký bởi một thực thể có thể được kiểm tra bởi bất kỳ thực thể nào quan tâm. Các thực thể này có thể an tâm rằng chỉ có chủ của khóa bí mật mới có thể tạo ra thông điệp này, bởi vì chỉ có người đó mới có khóa bí mật.



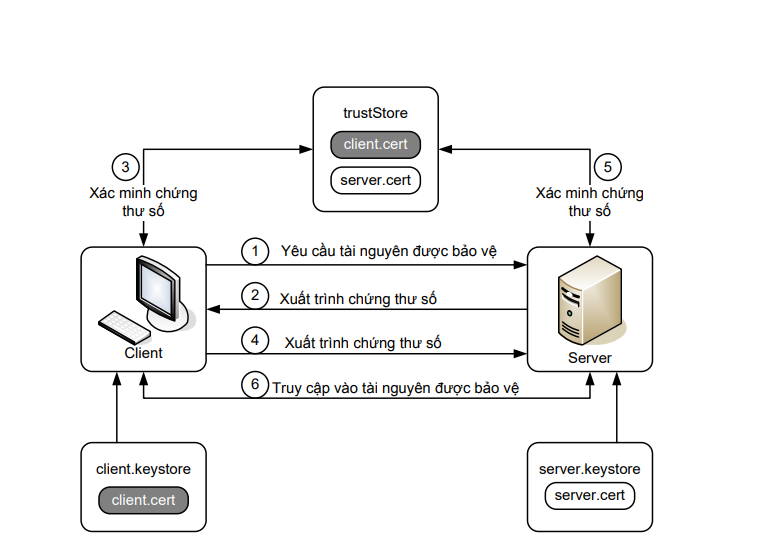
Hình 1.19. Ứng dụng xác thực

Các kỹ thuật xác thực

* Kỹ thuật xác thực độ tươi của thông điệp và tính sống động của người dùng: hỏi/đáp dấu thời gian,…



Hình 1.20. Xác thực từ xa dựa trên khoá công khai

* Kỹ thuật xác thực lẫn nhau và kỹ thuật xác thực sử dụng bên thứ ba tin cậy (Kerberos, PKI)

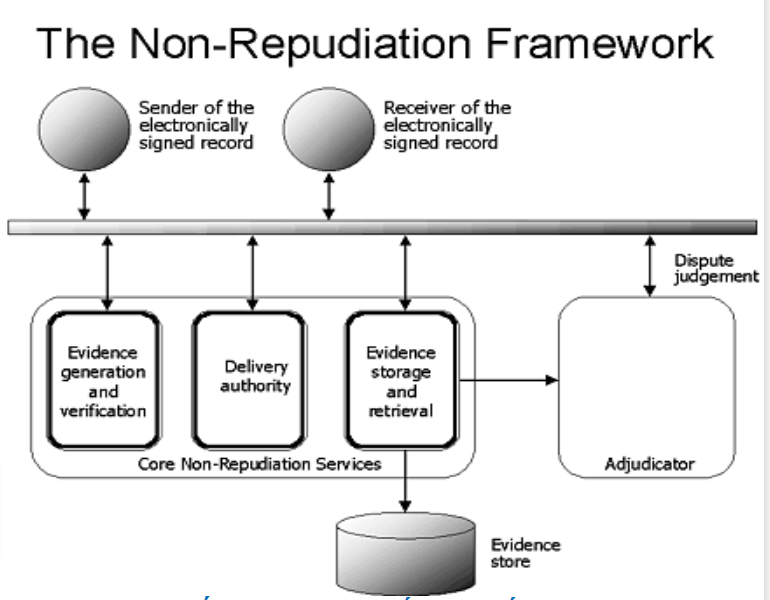
Hình 1.21. Xác thực dựa trên bên thứ 3 tin cậy

* Kỹ thuật trao đổi khóa: trao đổi khóa Diffie-Hellman

Chống chối bỏ:

Chối bỏ được định nghĩa là sự không thừa nhận của một trong các thực thể tham gia truyền thông rằng, anh ta không tham gia tất cả hoặc một phần cuộc truyền thông.

Chống chối bỏ nghĩa là đảm bảo dữ liệu không thể bị không thừa nhận hoặc giao tác bị từ chối. Đây là một dịch vụ bảo mật then chốt của bất kỳ ứng dụng thương mại nào trong đó việc trao đổi giá trị hay các quy định pháp luật được thỏa hiệp. Tính không thể chối từ được cung cấp thông qua mã hóa khóa công khai bằng chữ ký số. Khi dữ liệu được ký theo cách mật mã học sử dụng khóa bí mật của cặp khóa, bất kỳ ai có thể truy cập khóa công khai của cặp khóa này đều có thể xác định rằng chỉ có chủ của cặp khóa mới có thể ký vào dữ liệu.



Hình 1.22. Ứng dụng chống chối bỏ

Ví dụ nếu B gửi một biên nhận điện tử đã được ký số tới A, khẳng định rằng anh ta đã nhận được một văn bản cụ thể từ A, anh ta sau này không thể chối bỏ rằng đã nhận văn bản và các biện minh có thể:

- Anh ta đưa khóa bí mật để ký một cách chủ ý cho bên thứ ba để cho phép khả năng chối bỏ biên nhận văn bản.

- Khoá bí mật để ký của anh ta đã bị lộ mà anh ta không biết (và anh ta đã sơ xuất một chút trong việc bảo vệ nó một cách đúng đắn).

Dịch vụ “chống chối bỏ biên nhận” đem lại cho A một đảm bảo nào đó rằng B đã nhận văn bản và xác nhận đã tiếp nhận bằng chữ ký số.

# **Công nghệ ký số, mã mật file PDF và xây dựng ứng dụng web**

* 1. **Ký số, mã mật file PDF**
     1. ***Giới thiệu tổng quan về PDF***
        1. Lịch sử

Việc phổ biến định dạng PDF trong thời gian đầu tương đối chậm. Những phiên bản đầu tiên của PDF không hỗ trợ siêu liên kết bên ngoài, làm giảm tính hữu dụng của nó trên web. Kích thước tập tin tăng lên so với văn bản thuần cũng có nghĩa là thời gian để tải xuống một tài liệu PDF sẽ lâu hơn, đây cũng là một vấn đề với những thiết bị kết nối vốn đã chậm thời đó.

Định dạng PDF được thay đổi nhiều lần khác nhau và hiện tại vẫn tiếp tục được phát triển. Dưới đây là 9 phiên bản ứng với các phiên bản của Acrobat.

* (1993) – PDF 1.0 / Acrobat 1.0
* (1994) – PDF 1.1 / Acrobat 2.0
* (1996) – PDF 1.2 / Acrobat 3.0
* (1999) – PDF 1.3 / Acrobat 4.0
* (2001) – PDF 1.4 / Acrobat 5.0
* (2003) – PDF 1.5 / Acrobat 6.0
* (2005) – PDF 1.6 / Acrobat 7.0
* (2006) – PDF 1.7 / Acrobat 8.0
* (2008) – PDF 1.7, Adobe Extension Level 3 / Acrobat 9.0
  + - 1. Định nghĩa

**PDF** là từ viết tắt của **Portable Document Format**, nghĩa tiếng Việt là tệp định dạng tài liệu di động, nó được phát hành bởi Adobe. Phần mở rộng của tệp PDF là .PDF. Tệp PDF có thể chứa được văn bản, hình ảnh, nút tương tác, các siêu liên kết (hyperlink), video hay phông chữ nhúng (embedded fonts),...

Một số phần mềm đọc và chỉnh sửa file PDF : **Adobe Reader, Foxit Reader, Adobe Acrobat, WPS Office**

Ưu điểm:

- Khả năng hiển thị cùng một nội dung và bố cục khi xem trên bất kỳ hệ điều hành, thiết bị, ứng dụng hay phần mềm.

- Cho phép tích hợp nhiều nội dung khác nhau như văn bản, hình ảnh và đồ họa vector, video, ảnh GIF, tệp âm thanh, mô hình 3D, siêu liên kết và các nút điều hướng,... trên cùng một tệp PDF và có thể được sắp xếp dưới dạng báo cáo, bảng trình bày hay danh mục.

- Thuật toán nén hình ảnh giúp tệp PDF sử dụng dung lượng lưu trữ thấp, giảm áp lực lưu trữ lên ổ cứng thiết bị.

- Tính năng bảo mật dữ liệu cao khi cho phép người dùng cài đặt chế độ cấm in, chỉnh sửa hoặc chữ ký điện tử để xác thực tài liệu.

Nhược điểm :

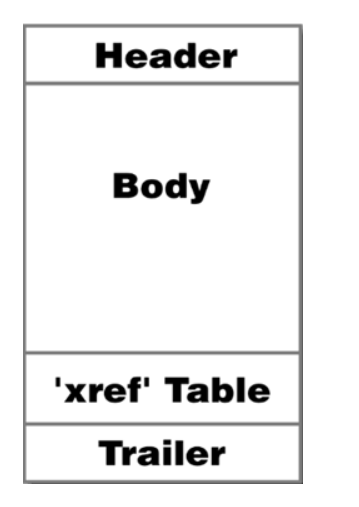
- Chi phí chỉnh sửa file PDF.

- File PDF được coi như là một hình ảnh vì thế khó thao tác với văn bản trong file.

- Nếu không có phần mềm chuyên dụng, bạn không thể đọc hay chỉnh sửa file PDF.

* + 1. ***Cấu trúc file PDF***

PDF có nhiều chức năng hơn chỉ là văn bản: nó có thể bao gồm hình ảnh và các phần tử đa phương tiện khác, được bảo vệ bằng mật khẩu, thực thi JavaScript, v.v. Cấu trúc cơ bản của tệp PDF được trình bày trong hình dưới đây:



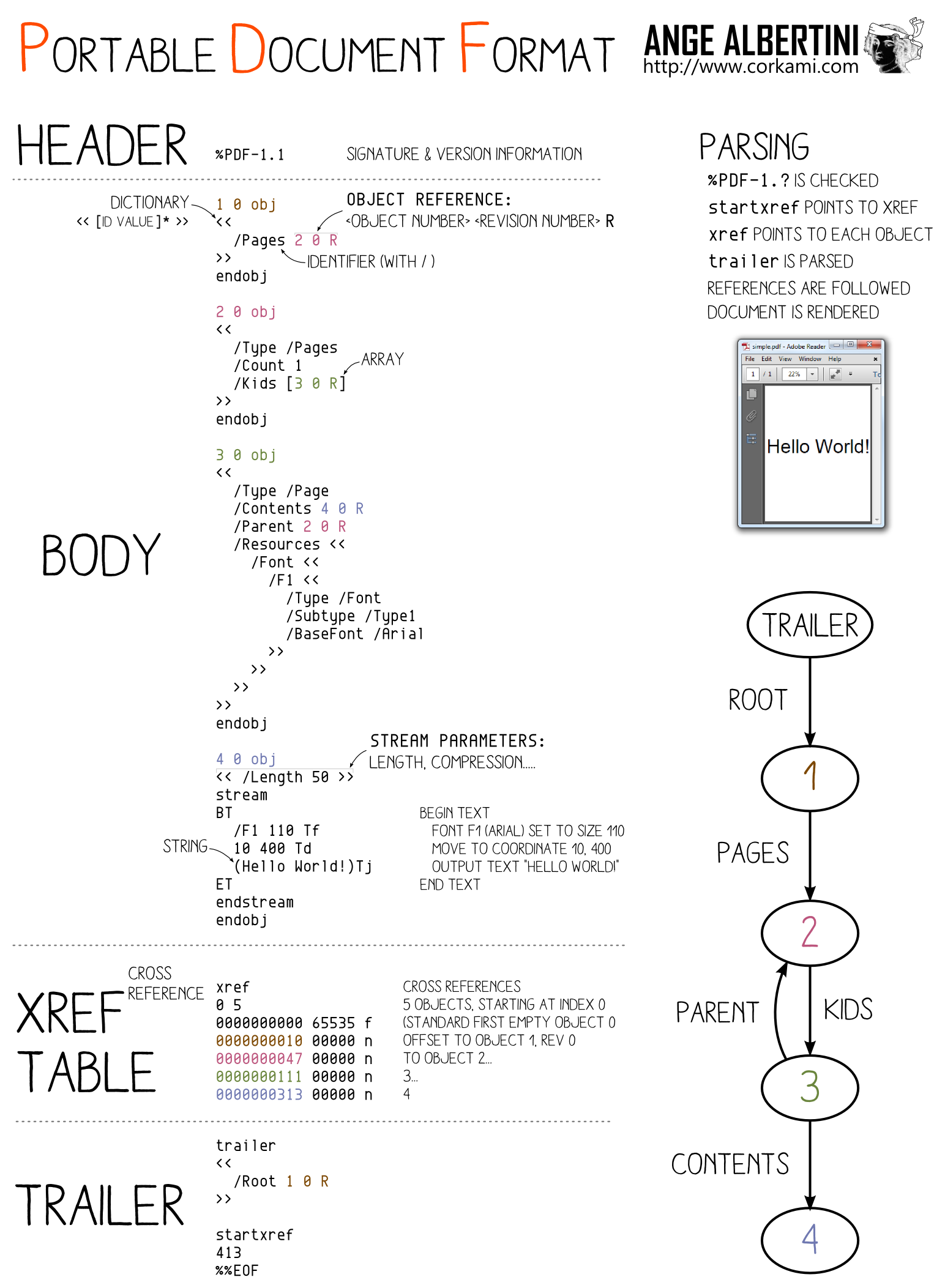
Hình 2.1. Cấu trúc cơ bản của tệp PDF

Header: Hiển thị phiên bản PDF

Body: Hiển thị nội dung chính mà người dùng có thể nhìn thấy

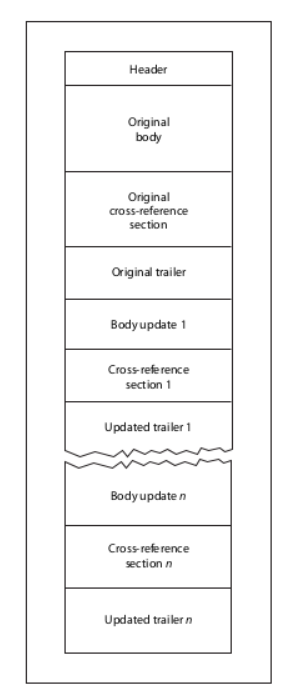
Xref table: Một thư mục liệt kê các đối tượng bên trong phần thân và vị trí của chúng (để hiển thị body);

Trailer: Nơi người xem PDF bắt đầu đọc tài liệu. Trailer chứa hai thông số quan trọng cho chương trình biết nơi bắt đầu xử lý tệp và nơi bắt đầu phần Xref.



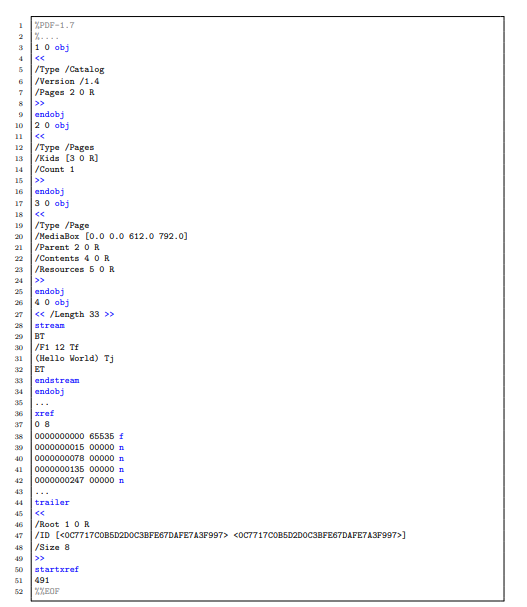
Hình 2.2. Chi tiết về cấu trúc file PDF

Tích hợp trong định dạng là một chức năng cập nhật.Ví dụ chúng ta chỉnh sửa một đoạn văn bản hay thêm vào hình vẽ, chú thích. Từ quan điểm kỹ thuật, chức năng này bổ sung thêm ba phần: cập nhật cho phần nội dung, thư mục Xref mới và trailer mới. Điều đó có hiệu quả giúp người dùng có thể thay đổi cách người dùng nhìn thấy các đối tượng và thêm nội dung mới.



Hình 2.3. Cấu trúc PDF sau khi được cập nhật gia tăng

Về bản chất, chữ ký điện tử cũng là một bản cập nhật gia tăng, thêm một phần tử khác và các phần tương ứng vào tệp.



Hình 2.4. Ví dụ về tệp PDF

* + 1. ***Chữ ký số trong file PDF***
       1. Chữ ký số

Chữ ký số : là thông điệp (có thể là văn bản, hình ảnh, hoặc video...) và được ký bằng khóa bí mật của người dùng nhằm mục đích xác định người chủ của thông điệp đó.

Mục đích của chữ ký số:

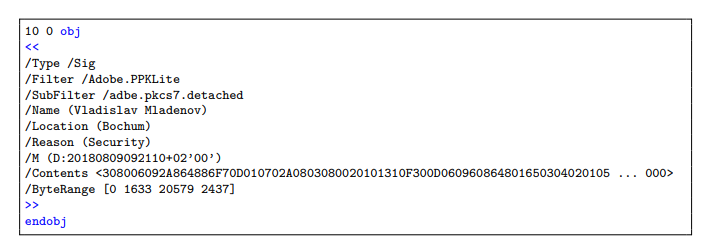
* Xác thực: xác định ai là chủ của thông điệp
* Tính toàn vẹn : kiểm tra xem thông điệp có bị thay đối
* Tính chống thoái thác: ngăn chặn việc người dùng từ chối việc tạo ra và gửi thông điệp

Chữ ký số trên PDF được mô tả chi tiết trong ISO 32000. ISO 32000 quy định rõ định định dạng PDF cho phép người dùng tin tưởng vào văn bản điện tử bằng cách hỗ trợ kiểm tra chữ ký số đảm bảo văn bản đã ký là toàn vẹn và của người ký. Định dạng cho phép nhiều chữ ký, hỗ trợ các thuật toán ký số thông dụng, cho phép hiển thị hình ảnh chữ ký tay thông thường. Các phiên bản PDF cũng như việc hỗ trợ chữ ký số trên PDF được cập nhật theo từng phiên bản. Phiên bản sau tương thích với phiên bản trước.

Kể từ phiên bản PDF 1.3, có thể thêm chữ ký điện tử trực tiếp vào tệp PDF bằng cách sử dụng trường biểu mẫu đặc biệt được gọi là “Trường chữ ký”. (Signature Field).

Trường biểu mẫu chữ ký tham chiếu đến cái gọi là “từ điển chữ ký “ (signature dictionary) chứa giá trị chữ ký và thông tin cần thiết khác, ví dụ:

mã hóa được sử dụng để lưu trữ giá trị chữ ký .



Hình 2.5. Từ điển chữ ký

Chữ ký sẽ được tạo ra bằng cách tính toán phần tóm tắt dữ liệu (hoặc một phần dữ liệu) trong một tài liệu và lưu trữ phần tóm tắt trong tài liệu. Để xác minh chữ ký, bản tóm tắt sẽ được tính toán lại và so sánh với bản được lưu trong tài liệu. Sự khác biệt trong các giá trị thông báo cho thấy rằng các sửa đổi đã được thực hiện kể từ khi tài liệu được ký.

Có hai kỹ thuật được xác định để tính toán chữ ký điện tử cho nội dung của toàn bộ hoặc một phần tệp PDF:

* byte range digest sẽ được tính trên một phạm vi byte trong tệp, điều này sẽ được chỉ ra bằng /ByteRange trong signature dictionary. Phạm vi này phải là toàn bộ tệp, bao gồm từ điển chữ ký nhưng loại trừ bản thân giá trị chữ ký (mục Content). Các phạm vi khác có thể được sử dụng nhưng vì chúng không kiểm tra tất cả các thay đổi đối với tài liệu, nên việc sử dụng chúng không được khuyến khích. Khi có bản tóm lược phạm vi byte, tất cả các giá trị trong từ điển chữ ký sẽ là các đối tượng trực tiếp.
* Ngoài ra, việc phát hiện sửa đổi có thể được chỉ định bởi một từ điển tham chiếu chữ ký. TransformMethod sẽ chỉ định phương pháp chung để phát hiện sửa đổi và TransformParams sẽ chỉ định các phần thay đổi của phương pháp.

Có hai loại chữ ký khác nhau:

* Chữ ký "chứng nhận" : Mỗi tài liệu chỉ được chứa một chữ ký xác nhận và chữ ký này phải là chữ ký đầu tiên trong tài liệu. Chữ ký chứng nhận cho phép tác giả của tài liệu hạn chế những thay đổi nào được phép áp dụng cho tài liệu sau khi anh ta tạo nó ban đầu (ví dụ: cho phép thêm các chữ ký hoặc chú thích khác vào tài liệu). Điều này đạt được bằng cách sử dụng phương pháp chuyển đổi "DocMDP".
* Chữ ký "phê duyệt": được sử dụng để phê duyệt các nội dung hiện tại của tài liệu; một tài liệu có thể chứa nhiều chữ ký phê duyệt. Tùy thuộc vào loại chữ ký mà các ứng dụng hiển thị thông tin khác nhau về chữ ký.

Transform methods: Phương thức chuyển đổi xác định đối tượng nào được đưa vào và đối tượng nào bị loại trừ khỏi tính toán tóm tắt đối tượng cần thiết để so sánh hai bản sửa đổi của tài liệu với nhau. Có bốn phương pháp biến đổi khác nhau được chỉ định:

* DocMDP: Phương pháp biến đổi “DocMDP” được sử dụng để phát hiện hoặc ngăn chặn các sửa đổi của tài liệu sau khi được tác giả ký. Nó được sử dụng bởi loại chữ ký chứng nhận; do đó, chỉ có một chữ ký sử dụng phương pháp biến đổi này trong một tài liệu và nó phải là chữ ký đầu tiên. Chữ ký bổ sung phải là chữ ký phê duyệt và sử dụng phương pháp biến đổi khác. Tác giả của tài liệu có thể sử dụng phương pháp biến đổi này để chỉ định những thay đổi nào được phép áp dụng cho tài liệu mà không làm mất hiệu lực chữ ký của anh ta. Điều này đạt được nhờ mục nhập /P trong từ điển tham số biến đổi: Nếu giá trị được đặt thành 1, tài liệu được coi là cuối cùng và bất kỳ thay đổi nào sẽ làm mất hiệu lực chữ ký; giá trị 2 cho phép người dùng khác điền vào các trường biểu mẫu và thêm (phê duyệt) chữ ký vào tài liệu; nếu giá trị được đặt thành 3 người dùng được phép tạo, sửa đổi và xóa chú thích.
* FieldMDP: Phương pháp biến đổi “FieldMDP” được sử dụng để phát hiện các thay đổi đối với giá trị của danh sách các trường biểu mẫu đã chỉ định. “FieldMDP” cho phép tác giả của tài liệu “khóa” các trường biểu mẫu nhất định và chỉ định các trường biểu mẫu nào được phép điền vào mà không làm mất hiệu lực chữ ký của tác giả.
* UR: Phương pháp chuyển đổi “UR” (“Quyền sử dụng”) được sử dụng để phát hiện những thay đổi có thể làm mất hiệu lực của “chữ ký quyền sử dụng”. Chữ ký này được sử dụng để kích hoạt “các tính năng tương tác bổ sung”.
* Identity: Phương thức biến đổi "Identity" được sử dụng khi tính toán một bản tóm tắt đối tượng mà không loại trừ bất kỳ đối tượng nào. Điều này có nghĩa là toàn bộ cây phân cấp đối tượng thay đổi và các thay đổi đối với bất kỳ đối tượng nào sẽ làm mất hiệu lực chữ ký.

|  |  |  |
| --- | --- | --- |
| Từ khóa | Kiểu dữ liệu | Giá trị |
| Type | name | (Tùy chọn) Loại đối tượng PDF mà từ điển này mô tả. Nếu có, sẽ là Sig cho một từ điển chữ ký. |
| Filter | name | (Bắt buộc) Tên của trình xử lý chữ ký được ưu tiên sử dụng khi xác thực chữ ký này. Nếu mục nhập Prop\_Build không có, nó cũng sẽ là tên của trình xử lý chữ ký đã được sử dụng để tạo chữ ký. Nếu Prop\_Build có mặt, nó có thể được sử dụng để xác định tên của trình xử lý đã tạo ra chữ ký. Một trình đọc phù hợp có thể thay thế một trình xử lý khác khi xác minh chữ ký, miễn là nó hỗ trợ định dạng SubFilter được chỉ định. Các trình xử lý chữ ký ví dụ là Adobe.PPKLite, Entrust.PPKEF, ClCl.Signlt. và VeriSign.PPKVS. |
| SubFilter | name | (Tùy chọn) Tên mô tả mã hóa giá trị chữ ký và thông tin khóa trong từ điển chữ ký. Người đọc có thể sử dụng bất kỳ trình xử lý nào hỗ trợ định dạng này để xác thực chữ ký.  (PDF 1.6) Các giá trị sẽ được sử dụng cho chữ ký mật mã khóa công khai: adbe.x509.rsa\_sha1, adbe.pkcs7.detached và adbe.pkcs7.sha1. Các giá trị khác có thể được xác định bởi các nhà phát triển và khi được sử dụng, sẽ có tiền tố là nhận dạng nhà phát triển đã đăng ký. |
| Contents | byte string | (Bắt buộc) Giá trị của chữ ký. Khi có ByteRange, giá trị phải là một chuỗi thập lục phân đại diện cho giá trị tóm lược byte range.  Đối với chữ ký khóa công khai, Nội dung phải là đối tượng dữ liệu nhị phân PKCS#1 được mã hóa DER hoặc đối tượng dữ liệu nhị phân PKCS#7 được mã hóa DER. |
| Cert | byte string hoặc array | ( Bắt buộc khi SubFilter là adbe.x509.rsa\_sha1) Một mảng các chuỗi byte sẽ đại diện cho chuỗi chứng chỉ X.509 được sử dụng khi ký và xác minh các chữ ký sử dụng mật mã khóa công khai hoặc một chuỗi byte nếu chuỗi chỉ có một mục nhập. Chứng chỉ ký sẽ xuất hiện đầu tiên trong mảng; nó sẽ được sử dụng để xác minh giá trị chữ ký trong Contents, và các chứng chỉ khác sẽ được sử dụng để xác minh tính xác thực của chứng chỉ ký.  Nếu SubFilter là adbe.pkcs7.detached hoặc adbe.pkcs7.sha1, mục nhập này sẽ không được sử dụng và chuỗi chứng chỉ sẽ được đặt trong PKCS#7 trong Contents. |
| ByteRange | array | (Bắt buộc đối với tất cả các chữ ký là một phần của trường chữ ký và chữ ký quyền sử dụng được tham chiếu từ mục nhập UR3 trong từ điển quyền) Một mảng các cặp số nguyên (độ lệch byte bắt đầu, độ dài tính bằng byte) sẽ mô tả phạm vi byte chính xác cho tính toán bản tóm lược. Nhiều dải byte không liền nhau sẽ được sử dụng để mô tả một bản tóm lược không bao gồm giá trị chữ ký (mục Contents). |
| Reference | array | (Tùy chọn; PDF 1.5) Một loạt các từ điển tham chiếu chữ ký. |
| Changes | array | (Tùy chọn) Một mảng ba số nguyên sẽ chỉ định các thay đổi đối với tài liệu đã được thực hiện giữa chữ ký trước đó và chữ ký này: theo thứ tự này, số trang được thay đổi, số trường được thay đổi và số trường được điền vào.  Thứ tự các chữ ký sẽ được xác định bởi giá trị của ByteRange. Vì mỗi chữ ký dẫn đến việc lưu tăng dần, các chữ ký sau này có giá trị độ dài lớn hơn. |
| Name | text string | (Tùy chọn) Tên của người hoặc cơ quan ký tài liệu. Giá trị này chỉ nên được sử dụng khi không thể trích xuất tên từ chữ ký. |
| M | date | (Tùy chọn) Thời điểm ký kết. Tùy thuộc vào trình xử lý chữ ký, đây có thể là thời gian máy tính chưa được xác minh thông thường hoặc thời gian được tạo theo cách có thể xác minh từ máy chủ thời gian an toàn.  Giá trị này chỉ nên được sử dụng khi thời điểm ký không có sẵn trong chữ ký.  Ví dụ: Dấu thời gian có thể được nhúng vào đối tượng dữ liệu nhị phân PKCS#7. |
| Location | text string | (Tùy chọn) Tên máy chủ CPU hoặc vị trí thực của việc ký kết. |
| Reason | text string | (Tùy chọn) Lý do ký kết, chẳng hạn như (Tôi đồng ý ...). |
| ContactInfo | text string | (Tùy chọn) Thông tin do người ký cung cấp để cho phép người nhận liên hệ với người ký để xác minh chữ ký. Ví dụ: Một số điện thoại. |
| R | interger | (Tùy chọn) Phiên bản của trình xử lý chữ ký được sử dụng để tạo chữ ký. (PDF 1.5) Mục nhập này sẽ không được sử dụng và thông tin sẽ được lưu trữ trong từ điển Prop Build. |
| V | interger | (Tùy chọn; PDF 1.5) Phiên bản của định dạng từ điển chữ ký. Nó tương ứng với việc sử dụng từ điển chữ ký trong ngữ cảnh giá trị của SubFilter. Giá trị là 1 nếu từ điển Tham chiếu được coi là quan trọng đối với việc xác nhận chữ ký. Giá trị mặc định: 0. |
| Prop\_Build | dictionary | (Tùy chọn; PDF 1.5) Một từ điển có thể được trình xử lý chữ ký sử dụng để ghi lại thông tin nắm bắt trạng thái của môi trường máy tính được sử dụng để ký, chẳng hạn như tên của trình xử lý được sử dụng để tạo chữ ký, ngày tạo phần mềm, phiên bản, và hệ điều hành. |
| Prop\_AuthTime | interger | (Tùy chọn; PDF 1.5) Số giây kể từ khi người ký được xác thực lần cuối, được sử dụng Trong các yêu cầu từ chối chữ ký. Nó nên được bỏ qua nếu giá trị không xác định. |
| Prop\_AuthType | name | (Tùy chọn; PDF 1.5) Phương pháp sẽ được sử dụng để xác thực người ký, được sử dụng trong các yêu cầu từ chối chữ ký. Các giá trị hợp lệ phải là mã PIN, Mật khẩu và Vân tay. |

Bảng 2.1. Các mục nhập của từ điển chữ ký

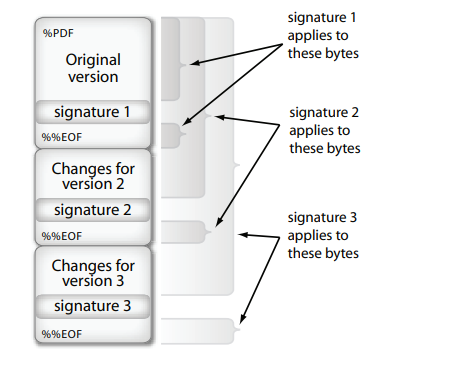
Các thuật toán có sẵn để tính toán thông báo và chữ ký cũng như độ dài có thể có của các khóa được sử dụng để ký phụ thuộc vào mã hóa ( /SubFilter trong từ điển chữ ký) ở trong sử dụng và phiên bản PDF mà tài liệu tuân thủ. Nói chung, SHA1 thuật toán thông báo và thuật toán ký RSA với các khóa có độ dài 1024 bit được hỗ trợ. Các thuật toán khác, ví dụ: thuật toán thông báo từ họ SHA2, Chữ ký RIPEMD160 hoặc DSA và các khóa dài hơn được hỗ trợ bởi một số mã hóa và một số phiên bản PDF cao hơn 1.3. Khóa RSA có độ dài 2048 hoặc 4096 bit.

|  |  |  |  |
| --- | --- | --- | --- |
| Giá trị trong SubFilter | | | |
|  | adbe.pkcs7.detached | adbe.pkcs7.sha1 | adbe.x509.rsa.sha1 |
| Message Digest | SHAl (PDF 1.3) SHA256 (PDF 1.6) SHA384 (PDF 1.7) SHA512 (PDF 1.7) RIPEMD160 (PDF 1.7) | SHAl (PDF 1.3) | SHAl (PDF 1.3) SHA256 (PDF 1.6) SHA384 (PDF 1.7) SHA512 (PDF 1.7) RIPHMD160 (PDF 1.7) |
| Thuật toán RSA | Lên đến 1024-bit (PDF 1.3)  Lên đến 2048-bit (PDF 1.5)  Lên đến 4096-bit (PDF 1.5) | Tương tự adbe.pkcs7.detached | Tương tự adbe.pkcs7.detached |
| Thuật toán DSA | Lên đến 4096-bit (PDF 1.) | Tương tự adbe.pkcs7.detached | Không hỗ trợ |

Bảng 2.2. Các thuật toán và độ dài khóa tối đa được hỗ trợ bởi các bảng mã và phiên bản PDF khác nhau.

Một số tài liệu có thể yêu cầu nhiều hơn một chữ ký.. Tuy nhiên, với PDF, bất kỳ sự thay đổi tài liệu bằng cách sửa đổi tệp (chẳng hạn như ký tên lại) sẽ làm mất hiệu lực chữ ký điện tử hiện có. Điều này là như vậy bởi vì giá trị băm được tính tại thời điểm xác minh sẽ không khớp với giá trị băm được mã hóa được tạo tại thời điểm ký. PDF giải quyết vấn đề này bằng cách hỗ trợ khả năng cập nhật gia tăng. Miễn là chữ ký bổ sung không bị ngăn chặn bởi các hạn chế quyền khác ,người ký chỉ có thể thêm trường chữ ký khác vào tài liệu và ký nó mà không làm mất hiệu lực chữ ký trước đó.

Các cập nhật gia tăng là minh bạch đối với người xem tài liệu, nhưng cho phép phát hiện và kiểm tra các sửa đổi đối với tệp. Tính năng này PDF nói chung và của riêng các tệp PDF đã ký, cho phép bất kỳ tệp PDF nào được sửa đổi bằng cách thêm thông tin sửa đổi vào cuối tệp trong phần cập nhật gia tăng. Không có bất kỳ thay đổi nào được yêu cầu đối với các byte đại diện cho phiên bản trước đó của tệp. Điều này cho phép thêm chữ ký bổ sung vào tệp PDF mà không sửa đổi bất kỳ dữ liệu nào được bao phủ bởi chữ ký trước đó.



Hình 2.6. Đa chữ ký và cập nhật gia tăng

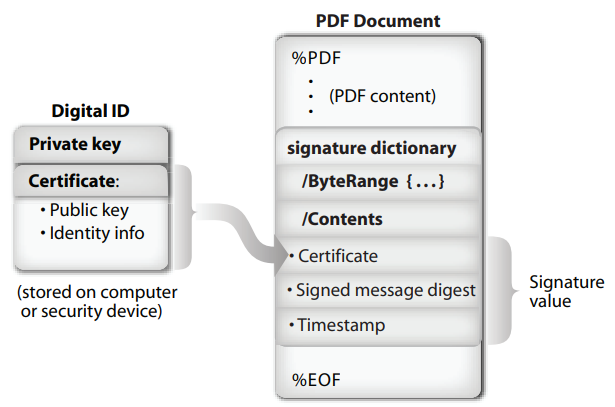
Phần đầu tiên là kết quả của ký lần 1. Khi ký tài liệu bằng chữ ký thứ hai, thay đổi nội dung và tạo một chữ ký mới. Chữ ký mới này dựa trên bản tóm lược thông điệp bao gồm toàn bộ mảng byte của bản ký lần 1. Kết quả là ký lần 2. Khi ký tài liệu bằng chữ ký thứ ba, tương tự lần 2.

Các chữ ký phải được được ký lần lượt , không được ký song song.

* + - 1. Tạo và kiểm tra chữ ký số

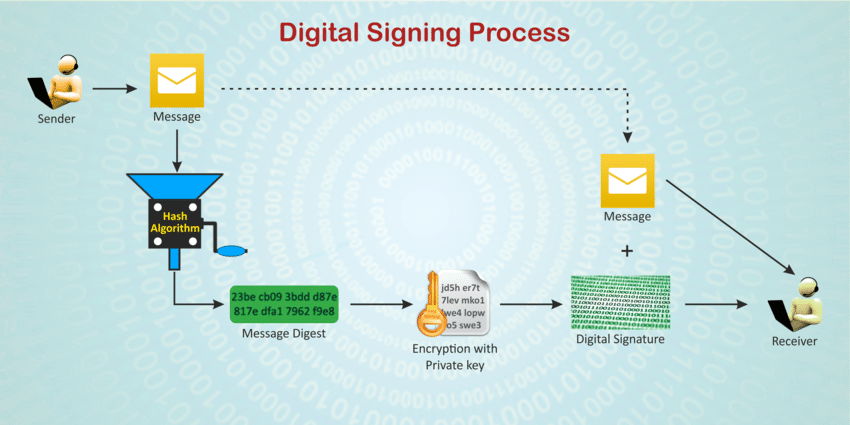
**Tạo chữ ký số**

Khi một tệp PDF được ký, chứng chỉ của người ký sẽ được nhúng vào tệp PDF. Giá trị chữ ký cũng có thể bao gồm thông tin bổ sung như hình ảnh chữ ký, dấu thời gian và dữ liệu khác có thể dành riêng cho người dùng, hệ thống hoặc ứng dụng.



Hình 2.7. Giá trị ByteRange và chữ ký

Sử dụng hàm băm để băm nội dung cần ký số. Sau đó sử dụng khóa bí mật và sử dụng thuật toán ký để mã hóa cùng nội dung đã được băm.



Hình 2.8. Quá trình ký số

Tài liệu được đọc dưới dạng luồng byte để thực hiện các hành động sau.

Các đối tượng mới và cập nhật được thêm vào tài liệu gốc: Đầu tiên, một danh mục tài liệu cập nhật (object 1 0) được thêm vào tài liệu. Nó bao gồm một từ điển AcroForm mới tham chiếu đến trường chữ ký mới (object8 0) và một đối tượng cần thiết cho sự xuất hiện của nó (object 9 0). Sau đó, hai đối tượng được thêm vào tài liệu. Giá trị của trường chữ ký (/V) là một tham chiếu đến từ điển chữ ký (object 10 0) cũng được thêm vào tài liệu. Nó chứa / Contents với độ dài cho giá trị chữ ký và /ByteRange. Bây giờ một trang mà tài liệu chứa đã được cập nhật (object 3 0) để thêm tham chiếu vào trường chữ ký. Sau đó, các đối tượng 11 0, 12 0, 13 0 và 14 0 được thêm vào tài liệu. Các đối tượng này xác định chi tiết về sự xuất hiện có thể nhìn thấy của chữ ký. Cuối cùng, một phần XRef mới (chứa các mục nhập cho tất cả các đối tượng mới và cập nhật) được tính toán và nối vào tệp cùng nhau với đoạn trailer được cập nhật và điểm đánh dấu %% EOF cuối cùng.

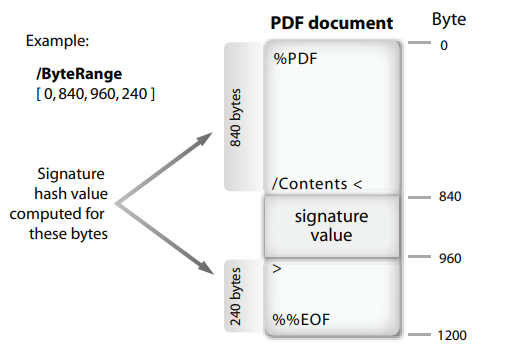
Sau khi tất cả các đối tượng cần thiết được thêm vào, có thể tính toán bù byte thực tế của / Contents. Để loại trừ giá trị của nó khỏi xác minh chữ ký, / ByteRange được cập nhật tương ứng.

Hàm băm trên các byte được chỉ định bởi / ByteRange được tính bằng cách sử dụng một trong các thuật toán có sẵn.

Giá trị băm được tính toán được ký bằng khóa riêng của người dùng và được mã hóa như được chỉ định bởi giá trị của / SubFilter. Tùy thuộc vào dữ liệu bổ sung mã hóa đã sử dụng (ví dụ: chứng chỉ của người ký, chuỗi chứng chỉ, thông tin thu hồi hoặc dấu thời gian) được mã hóa cùng với giá trị chữ ký.

Cuối cùng, dữ liệu được mã hóa được đặt trong /Contents (dưới dạng chuỗi thập lục phân ).

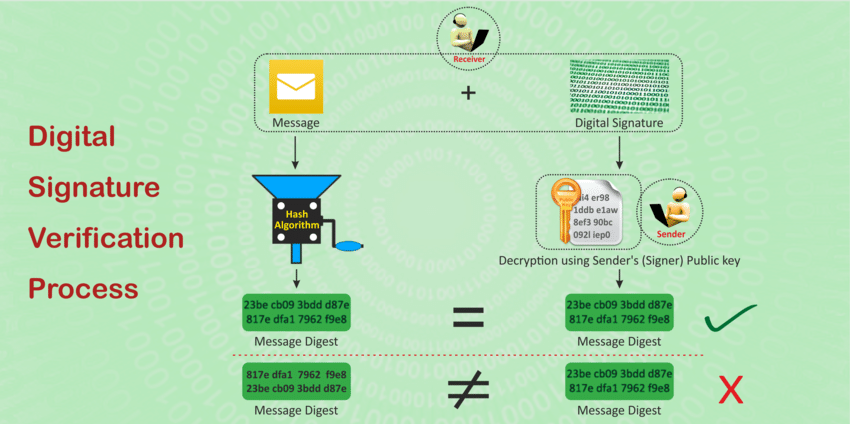
ByteRange là một mảng bốn số. Số đầu tiên trong mỗi cặp là phần bù trong tệp (từ đầu, bắt đầu từ 0) của dòng byte bắt đầu được đưa vào hàm băm. Số thứ hai là độ dài của luồng đó. Hai cặp xác định hai chuỗi byte xác định những gì sẽ được băm. Giá trị chữ ký thực tế được lưu trữ trong khóa /Contents giữa phần cuối của dãy thứ nhất và phần đầu của dãy thứ hai



Hình 2.9. Giá trị ByteRange và chữ ký

**Kiểm tra chữ ký số**

Kiểm tra chữ ký số ngược lại với việc ký dữ liệu. Việc kiểm tra chữ ký số sẽ cho bạn biết liệu dữ liệu đã ký có thay đổi hay không. Khi chữ ký điện tử được xác minh, chữ ký sẽ được giải mã bằng khóa công khai để tạo ra giá trị băm ban đầu. Dữ liệu đã được ký sẽ được băm. Nếu hai giá trị băm khớp nhau, thì chữ ký đã được xác minh.



Hình 2.10. Quá trình xác thực chữ ký số

Việc xác minh chữ ký điện tử trong tệp PDF được thực hiện ở chế độ nền khi người dùng mở tệp PDF đã ký bằng ứng dụng xử lý hỗ trợ chữ ký số. Mặc dù các chi tiết của quá trình xác minh phụ thuộc vào ứng dụng xử lý cụ thể, nhưng cấu trúc chung là tương tự cho tất cả các ứng dụng xử lý.

Đầu tiên giá trị /ByteRange được sử dụng để tính toán giá trị băm trên tài liệu ngoại trừ giá trị của /Content. Giá trị băm được tính toán được so sánh với giá trị được lưu trữ trong chữ ký. Nếu các giá trị khác nhau, quá trình xác minh được tiếp tục nhưng người dùng được thông báo rằng tài liệu đã bị thay đổi.

Chuỗi chứng chỉ được xây dựng để đảm bảo rằng chứng chỉ của người ký được tin cậy. Phải có ít nhất một đường dẫn từ chứng chỉ đến một nút tin cậy.

Thông tin thu hồi đối với chứng chỉ của người ký và các chứng chỉ trong chuỗi chứng chỉ được kiểm tra để đảm bảo rằng các chứng chỉ còn hiệu lực tại thời điểm tài liệu được ký. Nếu cả hai thông tin để sử dụng Giao thức trạng thái chứng chỉ trực tuyến (OCSP) và Danh sách Thu hồi Chứng chỉ (CRL) có sẵn, Acrobat cố gắng sử dụng OCSP trước.

Nếu chữ ký chứa dấu thời gian từ một máy chủ đáng tin cậy thì chứng chỉ cho dấu thời gian được xác minh theo cách giống như các chứng chỉ được mô tả ở trên, bao gồm cả việc kiểm tra thông tin thu hồi của nó.

* + - 1. Chuẩn chữ ký số file PDF ISO 32000-1 và PadES

**Chuẩn ISO 32000-1**

Tiêu chuẩn ISO cho "PDF đầy đủ chức năng" được xuất bản dưới hình thức tiêu chuẩn ISO 32000. Đặc tả chức năng đầy đủ của không chỉ là một tập hợp con của đặc điểm kỹ thuật Adobe PDF, trong khuyến nghị của ISO 32000-1 các chức năng PDF đầy đủ bao gồm tất cả mọi thứ được định nghĩa trong PDF 1.7.

ISO 32000-1: 2008 quy định cụ thể một dạng kỹ thuật số cho đại diện cho tài liệu điện tử cho phép người dùng trao đổi và xem các tài liệu điện tử độc lập với môi trường mà họ đã tạo ra trong hoặc môi trường mà họ đang xem hoặc in trong này dành cho các nhà phát triển phần mềm đó. tạo ra các tập tin PDF (phù hợp với các nhà văn), phần mềm đọc file PDF hiện có và giải thích nội dung của họ để hiển thị và tương tác (phù hợp với độc giả) và các sản phẩm PDF có thể đọc và / hoặc viết các tập tin PDF cho nhiều mục đích khác (phù hợp với sản phẩm).

**Chuẩn PadES**

Vào tháng 4 năm 2016, ETSI đã xuất bản một tiêu chuẩn mới (được cập nhật một) cho chữ ký nâng cao cho các tài liệu PDF đáp ứng các yêu cầu của Chỉ thị Châu Âu về Khung cộng đồng cho Chữ ký Điện tử (Chỉ thị 1999/93 / EC). Tiêu chuẩn mới là kết quả của việc điều chỉnh và mở rộng cho ISO 32000-1 do ETSI phối hợp với các chuyên gia PDF thực hiện. Tiêu chuẩn được đặt tên là PadES ( PDF Advanced Electronic Signature )

Tiêu chuẩn PaAdES được thiết lập dựa trên tiêu chí cho phép các tài liệu điện tử đã ký số có thể lưu trữ, tra cứu và sử dụng được sau hàng thập kỷ. Tại bất kỳ thời điểm nào trong tương lai, dù công nghệ có thay đổi với những bước tiến vượt bậc, người dùng vẫn phải xác thực được tài liệu điện tử – đặc biệt đối với hiệu lực của chữ ký số tại thời điểm ký. Đây chính là tiền đề cho sự ra đời của công nghệ “Ký số xác thực lâu dài”, còn gọi là Long-term Validation (LTV).

Với công nghệ Ký số xác thực lâu dài LTV, mốc thời gian ký số được ghi lại và lưu trữ ngay trong tài liệu điện tử. Trạng thái kích hoạt LTV được hiển thị trong các tham số của chữ ký số. Chứng cứ xác thực được lưu trữ ngay trong tài liệu điện tử, cho phép kiểm tra hiệu lực của chứng từ điện tử trong tương lai mà không phục thuộc vào hiệu lực hay trạng thái bị thu hồi của chứng thư số sau khi ký. Chính vì các chứng cứ này được lưu bên trong tài liệu đã ký số nên chúng cũng được xác thực và bảo vệ bởi chữ ký số của tài liệu điện tử, giảm thiểu tối đa các rủi ro sai sót hoặc giả mạo. Công nghệ LTV là lựa chọn hữu ích giúp cá nhân, tổ chức, doanh nghiệp hạn chế sự phụ thuộc vào hệ thống bên thứ ba và giảm thiểu những nghi vấn trong tương lai đối với các chứng thư số đã hết hạn/bị thu hồi.

LTV cung cấp hồ sơ về trạng thái của chứng thư số tại thời điểm ký.

Để xác thực chứng chỉ, cần phải biết ngày giờ mà tài liệu PDF được ký. Để điều này là không thể chối bỏ, chữ ký phải được đánh dấu bởi nhãn thời gian(timestamp).

Sự tiến bộ về công nghệ ký số lên tệp PDF thực sự hữu dụng nhờ các tính năng mới được hỗ trợ. Hiện nay, các Profiles ký số PAdES được cung cấp bởi Foxit PDF SDK bao gồm:

* **PAdES B: Basic Signature** – chữ ký PKCS#7 (adbe.pkcs7.detached) với tùy chọn sử dụng TimeStamp và thông điệp phản hồi OCSP được quy định tại tiêu chuẩn ISO-3200 PDF 1.7
* **PadES BES:Basic Enhanced Signatures** – Chữ ký CAdES (ETSI.CAdES.detached) với tùy chọn sử dụng TimeStamp và các phương thức bảo vệ bổ sung
* **PAdES EPES: Explicit Policy Enhanced Signatures** – Chữ ký CAdES (ETSI.CAdES.detached) với thuộc tính signature-policy-identifier
* **PAdES LTV: Long-Term Validation** – Chữ ký CAdES (ETSI.CAdES.detached) với TimeStamp, chứng thư số của CA và thông điệp phản hồi OCSP được đính kèm thời điểm ký. Các dấu thời gian TimeStamp và dữ liệu xác thực có thể được bổ sung phục vụ lưu trữ xác thực lâu dài.
  + - 1. Các thuật toán chữ ký số thông dụng

**Thuật toán chữ ký số RSA**

Phương pháp chữ ký số RSA được xây dựng dựa trên thuật toán mã hóa khóa công khai RSA

Để tạo một cặp khóa, RSA thực hiện các bước sau:

• Chọn 2 số nguyên tố lớn ngẫu nhiên 𝑝, 𝑞. Nhằm có sự an toàn tối đa nên chọn 𝑝 và 𝑞 có độ dài bằng nhau.

• Tính 𝑛 = 𝑝𝑞 và 𝜑 = (𝑝 − 1)(𝑞 − 1).

• Chọn ngẫu nhiên một số nguyên 𝑒 (1 < 𝑒 < 𝜑) sao cho 𝑔𝑐𝑑(𝑒,𝜑) = 1 với 𝑔𝑐𝑑 là ước số chung lớn nhất.

• Tính: 𝑑 = 𝑒 − 1 𝑚𝑜𝑑 𝜑. Kết quả là ta có được cặp khóa: khóa công khai (𝑛, 𝑒) và khóa bí mật (𝑛, 𝑑). Hai người sẽ sử dụng chung một hàm băm ℋ an toàn trước hiện tượng xung đột.

Để ký một thông điệp 𝑚, người ký thực hiện các bước sau:

• Dùng hàm băm ℋ để băm thông điệp 𝑚: = ℋ(𝑚).

• Sử dụng khóa bí mật (𝑛, 𝑑) để tính: 𝑠 = hd 𝑚𝑜𝑑 𝑛. Chữ ký của 𝑚 là 𝑠 và được gửi kèm với thông điệp 𝑚 đến người nhận.

Để xác nhận chữ ký, người nhận thực hiện các bước sau:

• Sử dụng khóa công khai (𝑛, 𝑒) của người ký để giải mã chữ ký: h = se 𝑚𝑜𝑑 𝑛.

• Sử dụng cùng hàm băm ℋ với người ký để băm thông điệp 𝑚: h ′ = 𝐻(𝑚).

• Chấp nhận chữ ký nếu h ′ = h. Ngược lại từ chối chữ ký.

**Thuật toán chữ ký số DSA**

Thuật toán chữ ký số DSA (Digital Signature Algorithm), là sự cải tiến của phương pháp ElGamal, được đề nghị bởi NIST vào tháng 8/1991 để sử dụng trong chuẩn chữ ký số DSS (Digital Signature Standard), được chỉ ra trong FIPS 186 , được chấp nhận năm 1993.

Việc tạo khóa gồm hai bước. Bước thứ nhất là lựa chọn các tham số cho thuật toán được chia sẻ giữa các người sử dụng khác nhau trong cùng hệ thống:

• Chọn một hàm băm mã hóa ℋ. Trong DSS chuẩn ℋ luôn là SHA-1, nhưng các hàm băm tốt hơn trong nhóm SHA cũng đang được sử dụng. Đôi khi đầu ra của một thuật toán băm mới hơn bị rút ngắn kích thước so với các thuật toán băm mới cũ để tương tích với cặp khóa hiện có.

• Chọn kích thước khóa 𝐿. Đây là thước đo chính quyết định sức mạnh mã hóa của khóa. DSS chuẩn ràng buộc 𝐿 là bội số của 64 và 512 ≤ 𝐿 ≤ 1024. Sau đó, FIPS 186-2 xác định 𝐿 luôn là 1024. Không lâu sau, NIST 800-57 đề nghị độ dài khóa là 2048 (hoặc 3072) để thời gian an toàn đến năm 2010 (hoặc 32 2030), sử dụng tương ứng với các giá trị băm và 𝑞 dài hơn. Bản thảo FIPS 186-3 [67] cũng tính đến các hàm băm sau này và các khóa dài hơn.

• Chọn một số nguyên tố 𝑞 cùng số bit với đầu ra của ℋ.

• Chọn một số nguyên tố 𝑝 độ dài 𝐿 bit sao cho 𝑝– 1 là bội của 𝑞. Tức là 𝑝 = 𝑞𝑧– 1 với số nguyên 𝑧 nào đó.

• Chọn 𝑔 = h(p-1)/q 𝑚𝑜𝑑 𝑝 với h bất kỳ (1 < h < 𝑝– 1), và chọn lại nếu kết quả là 1. Hầu hết cách chọn h đều nhận được g có thể sử dụng, thông thường chọn h = 2.

Các tham số thuật toán (𝑝, 𝑞, 𝑔) có thể chia sẻ giữa những người khác nhau trong hệ thống. Bước thứ hai tính các khóa bí mật và công khai của từng người:

• Chọn 𝑥 ngẫu nhiên sao cho 0 < 𝑥 < 𝑞.

• Tính 𝑦 = g𝑥 𝑚𝑜𝑑 𝑝.

• Khóa công khai là 𝑝, 𝑞, 𝑔, 𝑦 , khóa bí mật là 𝑥.

Phiên bản FIPS 186-3 sắp tới sử dụng SHA-224/256/384/512 là các hàm băm, kích thước của 𝑞 là 224 (hoặc 256 bit), và 𝐿 bằng 2048 (hoặc 3072).

Để ký một thông điệp 𝑚, người ký thực hiện các bước sau:

• Phát sinh một số ngẫu nhiêu 𝑘 (0 < 𝑘 < 𝑞) cho mỗi thông điệp.

• Tính 𝑟 = (𝑔k 𝑚𝑜𝑑 𝑝) 𝑚𝑜𝑑 𝑞.

• Tính 𝑠 = 𝑘-1 (ℋ(𝑚) + 𝑥𝑟)) 𝑚𝑜𝑑 𝑞.

• Tính toán lại chữ ký trong trường hợp không chắc chắn 𝑟 = 0 hoặc 𝑠 = 0.

• Chữ ký là (𝑟, 𝑠).

Để xác nhận chữ ký, người nhận thực hiện các bước sau:

• Loại bỏ chữ ký nếu 0 < 𝑟 < 𝑞 hoặc 0 < 𝑠 < 𝑞 không thỏa mãn.

• Tính 𝑤 = 𝑠-1 𝑚𝑜𝑑 𝑞.

• Tính 𝑢1 = (ℋ(𝑚) × 𝑤) 𝑚𝑜𝑑 𝑞.

• Tính 𝑢2 = (𝑟 × 𝑤) 𝑚𝑜𝑑 𝑞.

• Tính 𝑣 = g𝑢1 ×y𝑢2 𝑚𝑜𝑑 𝑝 𝑚𝑜𝑑 𝑞.

• Chữ ký có hiệu lực nếu 𝑣 = 𝑟.

Độ an toàn của thuật toán ElGamal phụ thuộc vào độ phức tạp của việc tìm lời giải cho bài toán logarit rời rạc nên cần phải sử dụng số nguyên tố 𝑝 đủ lớn (tối thiểu là 512 bit). Nếu số nguyên tố 𝑝 dài 512 bit thì chữ ký điện tử tạo ra có độ dài là 1024 bit và không phù hợp với các ứng dụng sử dụng thẻ thông minh vốn có nhu cầu sử dụng chữ ký ngắn. Phương pháp DSA đã giải quyết vấn đề này bằng cách sử dụng chữ ký 320 bit cho văn bản 160 bit với các phép tính được thực hiện trên tập con có 2 160 phần tử với 𝑝 là số nguyên tố 512 bit.

* + - 1. Các thuật toán hàm băm mật mã

**Thuật toán hàm băm MD5**MD5 (Message-Digest algorithm 5) là một hàm băm mật mã được sử dụng phổ biến, được thiết kể bởi Giáo sư Ronald L. Rivest tại trường MIT vào năm 1991 để thay thế cho hàm băm trước đó là MD4 (1990). Là một chuẩn Internet (RFC 1321), MD5 đã được dùng trong nhiều ứng dụng bảo mật và cũng được dùng phổ biến để kiểm tra tính toàn vẹn của tập tin. Cũng như các hàm băm khác như MD4 và SHS (Secure Hash Standard), MD5 là phương pháp có ưu điểm tốc độ xử lý rất nhanh, thích hợp với các thông điệp dài cho và cho ra giá trị băm dài 128 bit.

**Một số hàm băm khác**

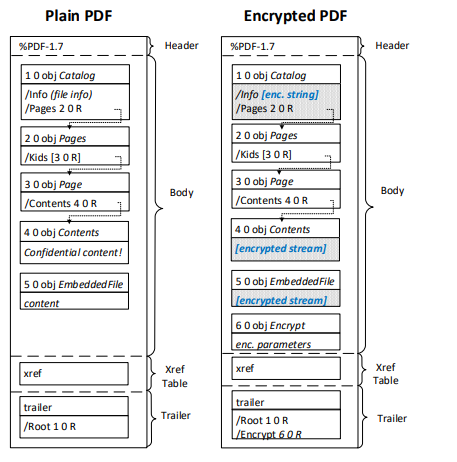
SHS là chuẩn gồm tập hợp các thuật toán băm mật mã an toàn (Secure Hash Algorithm – SHA) như SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 do NIST và NSA xây dựng.

Phương pháp SHA-1 (cũng như SHA-0) được xây dựng trên cùng cơ sở với phương pháp MD5. Tuy nhiên, phương pháp SHA-1 sử dụng trên hệ thống Big-endian thay vì Little-endiannhư phương pháp MD5. Hàm băm SHA-1 tạo ra thông điệp rút gọn kết quả có độ dài 160 bit nên thường được sử dụng.

Tất cả các thuật toán băm SHA là những hàm băm một chiều có thể xử lý thông điệp để tạo ra Message Digest. Những giải thuật này đảm bảo tính toàn vẹn của thông điệp: bất kỳ sự thay đổi nào trong thông điệp gốc sẽ có khả năng phát hiện ra sự thay đổi đó. Điều này rất có ý nghĩa trong vấn đề tạo và xác minh chữ ký số. Mỗi thuật toán SHA đều trải qua 2 quá trình đó là tiền xử lý và tính toán giá trị băm. Tiền xử lý bao gồm quá trình: thêm bit, chia thông điệp sau khi thêm bit thành N blocks m-bits, khởi tạo một giá trị cho việc tính giá trị băm. Tiếp theo là quá trình tính toán để tìm giá trị băm thông qua một quy trình lặp

* + 1. ***Mã mật file PDF***
       1. Giới thiệu về mã mật file PDF

Để đảm bảo tính bảo mật, tiêu chuẩn PDF xác định các chức năng mã hóa dành riêng cho PDF. Điều này cho phép bảo mật chuyển và lưu trữ các tài liệu nhạy cảm mà không cần thêm bất kỳ cơ chế bảo vệ. Sự quản lý chính giữa người gửi và người nhận có thể dựa trên mật khẩu (người nhận phải biết mật khẩu được sử dụng bởi người gửi hoặc nó phải được chuyển đến anh ta thông qua một kênh bảo mật) hoặc dựa trên khóa công khai (tức là người gửi biết chứng chỉ X.509 của người nhận).



Hình 2.11. Cấu trúc file PDF trước và sau mã hóa

Ta thấy rằng tài liệu PDF được mã hóa có cấu trúc bên trong giống như không được mã hóa. Tuy nhiên có hai điểm khác biệt chính giữa cả hai tệp:

* Đoạn trailer có bổ sung thêm mục Encrypt dictionary (từ điển mã hóa) ,báo hiệu cho người xem PDF rằng tài liệu đã được mã hóa và chứa thông tin cần thiết để giải mã nó.
* Theo mặc định, tất cả các chuỗi và luồng trong tài liệu là được mã hóa, ví dụ: 4 0 obj.

Encrypt dictionary là nơi lưu thông tin cần thiết để giải mã tài liệu. Nó chỉ định các thuật toán mật mã được sử dụng và quyền của người dùng,sẽ là giá trị của Encrypt trong trailer dictionary. Nếu không có mục này có nghĩa là không được mã hóa.

Filter của từ điển mã hóa xác định trình xử lý bảo mật của tệp, một mô-đun phần mềm thực hiện các khía cạnh khác nhau của quá trình mã hóa và kiểm soát quyền truy cập vào nội dung của tài liệu được mã hóa. PDF chỉ định một trình xử lý bảo mật dựa trên mật khẩu tiêu chuẩn mà tất cả các trình đọc tuân thủ sẽ hỗ trợ, nhưng những trình đọc tuân thủ có thể tùy chọn cung cấp các trình xử lý bảo mật bổ sung của riêng họ.

SubFilter chỉ định cú pháp của nội dung từ điển mã hóa. Nó cho phép khả năng tương tác giữa các trình xử lý. Nghĩa là một tài liệu có thể được giải mã bởi một trình xử lý khác với trình xử lý được ưu tiên nếu cả hai đều hỗ trợ định dạng được chỉ định bởi SubFilter.

V, trong việc chỉ định thuật toán nào sẽ sử dụng, xác định độ dài của khóa mã hóa, trên đó phải dựa trên mã hóa (và giải mã) dữ liệu trong tệp PDF. Đối với giá trị V 2 và 3, Length chỉ định độ dài chính xác của khóa mã hóa. Trong PDF 1.5, V 4 cho phép trình xử lý bảo mật sử dụng các thuật toán mã hóa và giải mã của riêng nó và để chỉ định các bộ lọc mã hóa để sử dụng trên các luồng cụ thể.

|  |  |  |
| --- | --- | --- |
| Mục nhập | Kiểu | Giá trị |
| Filter | name | (Bắt buộc) Tên của trình xử lý bảo mật ưu tiên cho tài liệu này. Nó sẽ là tên của trình xử lý bảo mật đã được sử dụng để mã hóa tài liệu. Nếu không có SubFilter, chỉ trình xử lý bảo mật này mới được sử dụng khi mở tài liệu. Nếu nó hiện diện, một trình đọc phù hợp có thể sử dụng bất kỳ trình xử lý bảo mật nào triển khai định dạng được chỉ định bởi SubFilter.  Tiêu chuẩn sẽ là tên của trình xử lý bảo mật dựa trên mật khẩu được tích hợp sẵn. |
| SubFilter | name | (Tùy chọn) Tên xác định hoàn toàn định dạng và cách giải thích nội dung của từ điển mã hóa. Nó cho phép các trình xử lý bảo mật khác với trình xử lý được Filter chỉ định giải mã tài liệu. Nếu mục này vắng mặt, các trình xử lý bảo mật khác sẽ không giải mã tài liệu. |
| V | number | (Tùy chọn) Mã chỉ định thuật toán được sử dụng để mã hóa và giải mã tài liệu:  0 - Một thuật toán không có tài liệu. Giá trị này sẽ không được sử dụng.  1 - "Thuật toán 1: Mã hóa dữ liệu bằng thuật toán RC4 hoặc AES", "Thuật toán mã hóa chung", với độ dài khóa mã hóa là 40 bit.  2 - (PDF 1.4) "Thuật toán 1: Mã hóa dữ liệu bằng thuật toán RC4 hoặc AES", "Thuật toán mã hóa chung", nhưng cho phép độ dài khóa mã hóa lớn hơn 40 bit.  3 - (PDF 1.4) Một thuật toán chưa được xuất bản cho phép độ dài khóa mã hóa nằm trong khoảng từ 40 đến 128 bit. Giá trị này sẽ không xuất hiện trong tệp PDF phù hợp.  4 - (PDF 1.5) Trình xử lý bảo mật xác định việc sử dụng mã hóa và giải mã trong tài liệu, sử dụng các quy tắc được chỉ định bởi các mục CF, StmF và StrF.  Giá trị mặc định nếu mục này bị bỏ qua sẽ là 0, nhưng khi có giá trị phải là giá trị 1 hoặc lớn hơn. |
| Length | interger | (Tùy chọn; chỉ khi V là 2 hoặc 3) Độ dài của khóa mã hóa, tính bằng bit. Giá trị phải là bội số của 8, trong khoảng 40 đến 128. Giá trị mặc định: 40. |
| CF | dictionary | (Tùy chọn; chỉ có nghĩa khi giá trị của V là 4) Một từ điển có khóa sẽ là tên bộ lọc mật mã và giá trị của nó sẽ là từ điển bộ lọc mật mã tương ứng. Mọi bộ lọc mật mã được sử dụng trong tài liệu phải có một mục nhập trong từ điển này, ngoại trừ các tên bộ lọc mật mã tiêu chuẩn. |
| StmF | name | (Tùy chọn; chỉ có ý nghĩa khi giá trị của V là 4; PDF 1.5) Tên của bộ lọc mật mã sẽ được sử dụng theo mặc định khi giải mã luồng. Tên phải là một khóa trong từ điển CF hoặc tên bộ lọc mật mã tiêu chuẩn được chỉ định trong Bảng 26. Tất cả các luồng trong tài liệu, ngoại trừ các luồng tham chiếu chéo (xem 7.5.8, "Các luồng tham chiếu chéo") hoặc các luồng có một mục nhập Mật mã trong mảng Bộ lọc của chúng (xem Bảng 6), sẽ được giải mã bởi trình xử lý bảo mật, sử dụng bộ lọc mật mã này.  Giá trị mặc định: Danh tính. |
| StrF | name | (Tùy chọn; chỉ có ý nghĩa khi giá trị của V là 4) Tên của bộ lọc mật mã sẽ được sử dụng khi giải mã tất cả các chuỗi trong tài liệu. Tên sẽ là một khóa trong từ điển CF hoặc một tên bộ lọc mật mã tiêu chuẩn.  Giá trị mặc định: Identity. |
| EFF | name | (Tùy chọn; chỉ có ý nghĩa khi giá trị của V là 4) Tên của bộ lọc mã hóa sẽ được sử dụng khi mã hóa các luồng tệp nhúng không có bộ xác định bộ lọc mã hóa riêng; nó phải tương ứng với một khóa trong từ điển CF hoặc tên bộ lọc mật mã tiêu chuẩn. |

Bảng 2.3. Các giá trị của V

Các thuật toán thường dùng

RC4 là một mật mã dòng đối xứng: cùng một thuật toán sẽ được sử dụng cho cả mã hóa và giải mã, và thuật toán không thay đổi độ dài của dữ liệu.

Thuật toán AES (Advanced Encryption Standard) là một khối đối xứng mật mã: cùng một thuật toán sẽ được sử dụng cho cả mã hóa và giải mã, và độ dài của dữ liệu khi mã hóa được làm tròn thành bội số của kích thước khối, được cố định luôn là 16 byte.

#### Từ điển mã hóa khóa công khai

Từ điển mã hóa cho trình xử lý bảo mật khóa công khai chứa các mục nhập phổ biến được hiển thị trong Bảng 2 có các giá trị được mô tả ở trên. Ngoài ra, chúng có thể chứa mục nhập được hiển thị trong Bảng như được mô tả dưới đây.

Mục nhập Filter sẽ là tên của trình xử lý bảo mật khóa công khai.

Ví dụ về các trình xử lý bảo mật hiện có hỗ trợ mã hóa khóa công khai là Entrust.PPKEF, Adobe.PPKLite và Adobe.PubSec. Trình xử lý này sẽ là trình xử lý ưu tiên khi mã hóa tài liệu.

Các giá trị được phép của mục SubFilter để sử dụng với các trình xử lý bảo mật khóa công khai phù hợp là adbe.pkcs7.s3, adbe.pkcs7.s4 sẽ được sử dụng khi không sử dụng bộ lọc mật mã và adbe.pkcs7.s5 sẽ được sử dụng khi sử dụng bộ lọc mật mã.

Các mục CF, StmF và StrF có thể xuất hiện khi SubFilter là adbe.pkcs7.s5.

|  |  |  |
| --- | --- | --- |
| Key | Type | Value |
| Recipients | Array | (Bắt buộc khi SubFilter là adbe.pkcs7.s3 hoặc adbe.pkcs7.s4; PDF 1.3) Một mảng chuỗi byte, trong đó mỗi chuỗi là một đối tượng PKCS # 7 liệt kê những người nhận đã được cấp quyền truy cập như nhau vào tài liệu. Dữ liệu chứa trong đối tượng PKCS # 7 sẽ bao gồm cả khóa mật mã được sử dụng để giải mã dữ liệu đã mã hóa và các quyền truy cập áp dụng cho danh sách người nhận. Sẽ chỉ có một đối tượng PKCS # 7 cho mỗi bộ quyền truy cập duy nhất; nếu một người nhận xuất hiện trong nhiều danh sách, các quyền được sử dụng sẽ là những quyền trong danh sách phù hợp đầu tiên. |
| P | interger | (Bắt buộc) Một tập hợp các cờ chỉ định các thao tác nào sẽ được phép khi tài liệu được mở với quyền truy cập của người dùng. Nếu bit 2 được đặt thành 1, tất cả các bit khác sẽ bị bỏ qua và mọi hoạt động đều được phép. Nếu bit 2 được đặt thành 0, quyền cho các hoạt động dựa trên giá trị của các cờ còn lại được xác định trong dưới đây |

Bảng 2.4. Các mục nhập từ điển mã hóa bổ sung cho các trình xử lý bảo mật khóa công khai

Giá trị của mục nhập P sẽ được hiểu là một số lượng 32 bit không dấu chứa một tập hợp các cờ chỉ định quyền truy cập nào sẽ được cấp khi tài liệu được mở bằng quyền truy cập của người dùng. Một bit 1 ở bất kỳ vị trí nào sẽ cho phép quyền truy cập tương ứng.

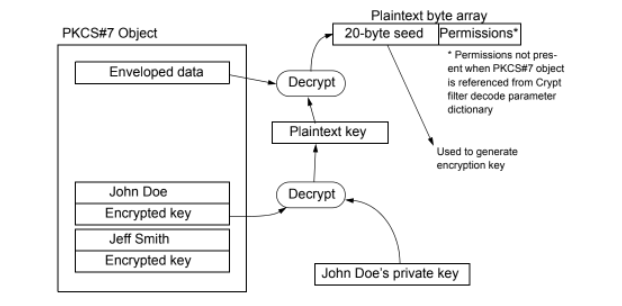
|  |  |
| --- | --- |
| Vị trí bit | Ý nghĩa |
| 2 | Khi thiết lập cho phép thay đổi mã hóa và bật tất cả các quyền khác. |
| 3 | In tài liệu (có thể không ở mức chất lượng cao nhất, tùy thuộc vào việc bit 12 có được đặt hay không). |
| 4 | Sửa đổi nội dung của tài liệu bằng các thao tác khác với các thao tác được điều khiển bởi bit 6, 9 và 11. |
| 5 | Sao chép hoặc trích xuất văn bản và đồ họa từ tài liệu bằng các thao tác khác với thao tác được điều khiển bởi bit 10. |
| 6 | Thêm hoặc sửa đổi chú thích văn bản, điền vào các trường biểu mẫu tương tác, và nếu bit 4 cũng được đặt, hãy tạo hoặc sửa đổi các trường biểu mẫu tương tác (bao gồm cả trường chữ ký). |
| 9 | Điền vào các trường biểu mẫu tương tác hiện có (bao gồm cả trường chữ ký), ngay cả khi bit 6 rõ ràng. |
| 10 | Trích xuất văn bản và đồ họa (để hỗ trợ người dùng khuyết tật hoặc cho các mục đích khác). |
| 11 | Lắp ráp tài liệu (chèn, xoay hoặc xóa các trang và tạo dấu trang hoặc hình ảnh thu nhỏ), ngay cả khi bit 4 rõ ràng. |
| 12 | In tài liệu ra bản trình bày mà từ đó có thể tạo ra một bản sao kỹ thuật số trung thực của nội dung PDF. Khi bit này rõ ràng (và bit 3 được đặt), việc in bị giới hạn ở mức độ thấp của hình thức, có thể chất lượng bị giảm sút. |

Bảng 2.5. Quyền truy cập của người dùng trình xử lý bảo mật khóa công khai

* + - 1. Mã mật hóa và giải mã mật

**Mã mật hóa**

Hình sau minh họa cách các đối tượng PKCS # 7 sẽ được sử dụng khi mã hóa tệp PDF. Đối tượng PKCS # 7 được thiết kế để đóng gói và mã hóa những gì được gọi là dữ liệu được bao bọc (enveloped data).



Hình 2.12. Public-key encryption algorithm

Dữ liệu được bao bọc trong đối tượng PKCS # 7 chứa vật liệu khóa sẽ được sử dụng để giải mã tài liệu. Một khóa sẽ được sử dụng để mã hóa (và giải mã) dữ liệu được bao bọc. Khóa này (khóa văn bản rõ) sẽ được mã hóa cho từng người nhận, sử dụng khóa công khai của người nhận đó và sẽ được lưu trữ trong đối tượng PKCS#7 (làm khóa được mã hóa cho từng người nhận). Để giải mã tài liệu, khóa đó sẽ được giải mã bằng khóa riêng của người nhận, khóa này mang lại khóa (bản rõ) được giải mã. Đến lượt mình, khóa đó sẽ được sử dụng để giải mã dữ liệu được bao bọc trong đối tượng PKCS#7, dẫn đến một mảng byte bao gồm các thông tin sau:

* Một hạt giống 20 byte sẽ được sử dụng để tạo khóa mã hóa được sử dụng bởi "Thuật toán 1: Mã hóa của dữ liệu sử dụng thuật toán RC4 hoặc AES ". Hạt giống sẽ là một số ngẫu nhiên duy nhất được tạo bởi trình xử lý bảo mật đã mã hóa tài liệu.
* Giá trị 4 byte xác định các quyền, byte ít quan trọng nhất đầu tiên.
* Khi SubFilter là adbe.pkcs7.s3, các quyền liên quan sẽ chỉ là những quyền được chỉ định cho bản sửa đổi 2 của trình xử lý bảo mật tiêu chuẩn.
* Đối với adbe.pkcs7.s4, các trình xử lý bảo mật của quyền của phiên bản 3 sẽ được áp dụng.
* Đối với adbe.pkcs7.s5, hỗ trợ việc sử dụng bộ lọc mật mã, các quyền sẽ giống như adbe.pkcs7.s4 khi bộ lọc mật mã được tham chiếu từ các mục StmF hoặc StrF của mã hóa từ điển. 4 byte quyền sẽ không có trong dữ liệu được bao bọc.
* Dữ liệu cần mã hóa là các đối tượng như chuỗi , stream trong body
* Các thuật toán sẽ được sử dụng để mã hóa dữ liệu là: RC4 với

độ dài khóa lên đến 256 bit, DES, Triple DES, RC2 với độ dài khóa lên đến 128 bit, 128 bit AES trong CBC, 192-bit AES ở chế độ CBC, 256-bit AES ở chế độ CBC.

Khóa mã hóa được sử dụng bởi thuật toán RC4 hoặc AES sẽ là

được tính bằng hàm băm SHA-1, theo thứ tự:

* 20 byte hạt giống.
* Các byte của mỗi mục trong mảng Recipients của các đối tượng PKCS # 7 theo thứ tự mà chúng xuất hiện trong mảng.
* 4 byte với giá trị OxFF nếu khóa đang được tạo nhằm mục đích sử dụng trong mã hóa cấp tài liệu và siêu dữ liệu tài liệu được để dưới dạng văn bản rõ.

n/8 byte đầu tiên của thông báo kết quả sẽ được sử dụng làm khóa mã hóa, trong đó n là độ dài bit của khóa mã hóa.

* Sử dụng mã khóa công khai của người nhận để mã hóa khóa mã hóa đó

**Giải mã mật**

Giải mã dữ liệu được mã hóa bởi trình xử lý bảo mật, tái tạo dữ liệu như trước khi mã hóa.

#### Trình xử lý bảo mật khóa công khai

Trình xử lý bảo mật có thể sử dụng công nghệ mã hóa khóa công khai để mã hóa tài liệu. Khi làm như vậy, việc chỉ định một hoặc nhiều danh sách người nhận, trong đó mỗi danh sách có các quyền truy cập duy nhất của riêng nó có thể được thực hiện. Chỉ những người nhận được chỉ định mới được phép mở tài liệu hoặc nội dung được mã hóa.

Trình xử lý bảo mật khóa công khai sử dụng cú pháp mã hóa nhị phân Tiêu chuẩn mã hóa khóa công khai số 7 (PKCS#7) để mã hóa danh sách người nhận, khóa giải mã và thông tin quyền truy cập.

Khi mã hóa dữ liệu, chứng chỉ khóa công khai X.509 của mỗi người nhận sẽ khả dụng. Khi giải mã dữ liệu, trình đọc phù hợp sẽ quét danh sách người nhận mà nội dung được mã hóa và sẽ cố gắng tìm sự trùng khớp với chứng chỉ thuộc về người dùng. Nếu tìm thấy khớp, người dùng yêu cầu quyền truy cập vào khóa cá nhân tương ứng, khóa này có thể yêu cầu xác thực, có thể sử dụng mật khẩu. Sau khi có được quyền truy cập, khóa riêng sẽ được sử dụng để giải mã dữ liệu đã mã hóa.

* 1. **Công nghệ xây dựng ứng dụng web** 
     1. ***Tổng quan về ứng dụng web***

Web là công nghệ trên Internet cho phép thể hiện các thông tin một cách sinh động, gần gũi với con ngời hơn trên 1 trang thông tin gọi là trang Web. Trang Web được trình bày trên các bộ duyệt Web (Browser) trên các máy Client. Trang Web chính là các file văn bản dạng Text được cấu trúc hoá theo ngôn ngữ HTML.

Ứng dụng web là một ứng dụng client/server ( máy chủ - máy khách) sử dụng giao thức HTTP để tương tác với người dùng hay hệ thống khác.Người dùng gửi và nhận các thông tin từ máy chủ WEB thông qua việc tác động vào các trang web.Ứng dụng web có thể là các trang tin tức, trao đổi mua bán, các diễn đàn, game online,..

* + 1. ***Công nghệ xây dựng ứng dụng Web***

Có thể coi ứng dụng có hai phần: frontend, tức là phần ứng dụng mà người dùng sử dụng để tương tác với các tài nguyên ứng dụng và backend, là nơi lưu trữ tất cả các tài nguyên bạn đã tạo.

Bạn có thể dùng các frontend khác nhau trong ứng dụng để mang đến trải nghiệm tốt nhất cho người dùng, cũng như tận dụng các tính năng độc đáo ở mỗi nền tảng. Đây cũng là nơi bạn xác định các quy tắc chi phối ứng dụng dựa trên các trường hợp sử dụng theo nghiệp vụ.

Trong đồ án này, phần Frontend sẽ được xây dựng dựa trên Angular Framwork và phần Backend sử dụng ASP.Net web API.

* + - 1. ASP.NET

ASP.NET là một nền tảng dành cho phát triển web, được Microsoft phát hành và cung cấp lần đầu tiên vào năm 2002. Nền tảng được sử dụng để tạo ra các ứng dụng web-based

Các ứng dụng ASP.Net có thể được viết bằng nhiều ngôn ngữ .Net khác nhau. Trong đó có các kiểu ngôn ngữ như C #, VB.Net và J #.

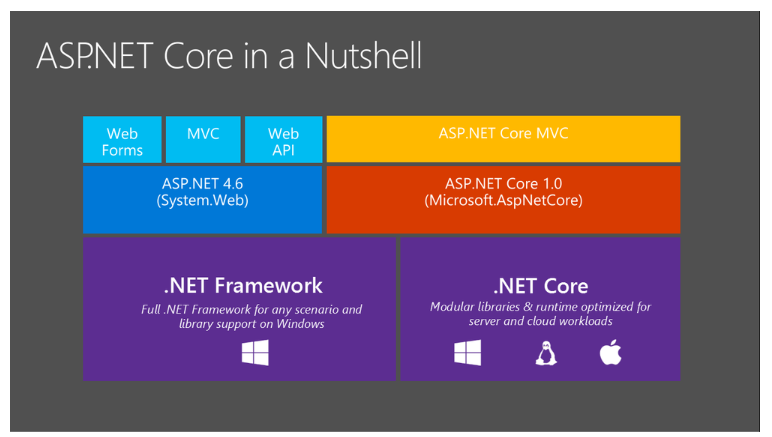


Hình 2.13. ASP.NET

ASP viết đầy đủ là Active Server Pages, và .NET là viết tắt của Network Enabled Technologies.

Cấu trúc thành phần của ASP.NET

Cấu trúc thành phần của **ASP.NET** bao gồm các yếu tố: Ngôn ngữ, thư viện và thời gian chạy CLR.



Ngôn ngữ: Là tập con của .NET Framework, ASP.NET sở hữu rất nhiều ngôn ngữ lập trình như C#, VB.net, PHP, JavaScript,... Trong đó C# và VB.net được sử dụng nhiều nhất trong phát triển ứng dụng web.

Thư viện**:**.ASP. NET Framework kế thừa các lớp thư viện từ .NET. Thư viện này được dùng rất phổ biến trong phát triển ứng dụng web.

Thời gian chạy CLR**:** Đây là một trong các cơ sở hạ tầng của phần đông các kiểu ngôn ngữ lập trình phổ thông. Trong các tình huống bình thường, CLR thực hiện các thao tác để xử lý các trường hợp đặc biệt và thu gom rác.

Đặc điểm của ASP.NET

**Code Behind Mode/ Trạng thái code rời**

Đây là khái niệm về tách rời thiết kế và mã code. Bằng cách tách rời như vậy, việc duy trì ứng dụng ASP.Net trở nên dễ dàng hơn. Loại file thông dụng của ASP.Net là aspx. Giả sử chúng ta có một trang web có tên là MyPage.aspx, sẽ có một tệp khác có tên là MyPage.aspx.cs biểu thị cho phần mã code của trang. Bởi vậy, Visual Studio mới tạo ra các tập tin riêng biệt cho mỗi trang web, một cho phần thiết kế và một dành cho mã code.

**State Management/Quản lý trạng thái**

ASP.Net có các phương tiện để kiểm soát quản lý trạng thái, trong khi HTTP được biết đến là một giao thức "không trạng thái". Lấy một ví dụ về ứng dụng giỏ hàng: khi một user đã chọn lựa xong những gì mình muốn mua và đưa ra đưa ra quyết định mua hàng trên trang web, người đó sẽ nhấn nút gửi.

Ứng dụng cần ghi nhớ các mục mà người dùng đã chọn mua. Đây là hành động ghi nhớ trạng thái của một ứng dụng tại một thời điểm trong hiện tại. Vì HTTP là giao thức không trạng thái nên khi user truy cập các web bán hàng, HTTP sẽ không lưu trữ thông tin trên các giỏ hàng.

**Caching – Bộ nhớ Cache**

ASP.Net cũng có thể thực hiện chức năng của Caching, qua đó cải thiện hiệu suất làm việc cho ứng dụng. Với việc lưu bộ nhớ đệm cache, các trang thường xuyên được người dùng yêu cầu có thể được lưu trữ ở một vị trí tạm thời. Các trang này có thể được truy xuất nhanh hơn và người dùng có thể nhận được các phản hồi tốt hơn. Nhờ vậy, bộ nhớ đệm có thể giúp cải thiện đáng kể hiệu suất của ứng dụng.

* + - 1. Angular

Angular được xe là một open source (mã nguồn mở) hay frameworks miễn phí chuyên dụng cho công việc thiết kế web. Angular được phát triển từ những năm 2009 và được duy trì bởi Google. Frameworks này được xem là frameworks front end mạnh mẽ nhất chuyên dụng bởi các lập trình viên cắt HTML cao cấp.

Angular được ứng dụng rộng rãi với mục đích xây dựng project Single Page Application (SPA). Hiện tại, Version stable của Angular là Angular 9 (released on February 7, 2020) với TypeScript 3.6 và 3.7.

Trên thực tế, Angular sở hữu hai phiên bản hoàn toàn khác nhau như sau:

* **Phiên bản 1:** Phiên bản này có tên chính thức là AngularJS đến hiện nay nó vẫn đang được phát triển hoàn thiện bởi các chuyên gia. Vào năm 2018, thì phiên bản 1.7.3 đã được nhà sản xuất phát hành. Đây là phiên bản được hoàn thành thực sự dựa trên Javascript bởi những lập trình viên viết mã theo ngôn ngữ Javascript thì nó thường dễ làm và dễ tìm kiếm Tuy nhiên, hiệu năng khi so sánh với phiên bản ReactJS thì vẫn còn nhiều công ty xây dựng phải cân nhắc sử dụng Angular với phiên bản khác 1.
* **Phiên bản 2:**Phiên bản này được gọi với cái tên chính thức là Angular. Nó sở hữu tư duy thay đổi hoàn toàn với phiên bản thứ 1 và mang lại tham vọng giúp cho Angular có thể đánh bại ReactJS. AngularJS được xem là một trong những framework sở hữu cấu trúc ứng dụng dạng động và cho phép bạn có thể sử dụng HTML như là ngôn ngữ mẫu cũng như mở việc mở rộng cú pháp của HTML.

Các đặc trưng cơ bản của Angular là gì?

* Được sử dụng để có thể phát triển dựa trên JavaScript.
* Angular có khả năng tạo các ứng dụng client-side dựa trên mô hình MVC.
* Angular sở hữu khả năng tương thích cao có thể tự động xử lý dễ dàng các mã Javascript sao cho phù hợp với các trình duyệt nhất.
* Khi có mã nguồn mở và miễn phí thì nó sẽ được sử dụng rộng rãi hơn.

Thông thường, kiến trúc của một ứng dụng Angular là dựa trên những ý tưởng liên quan đến Components. Mỗi một ứng dụng Angular thường bắt đầu với những level trên cùng tên gọi là Root Component.

**Cách thức hoạt động của Angular**

Sau khi AngularJS được nhúng vào trang thì nó sẽ hiển thị lên cho việc phân tích các mã lệnh HTML. Mã lệnh HTML này sẽ có thẻ với thuộc tính ng-app=””. Khi đó, thuộc tính này sẽ được sử dụng để bắt đầu cho việc khởi tạo ứng dụng AngularJS. Thẻ tiếp theo có thuộc tính ng-model=”name” giúp tạo ra biến name bên trong ứng dụng AngularJS trên. Từ đó, giá trị của biến luôn bằng với giá trị của trường cuối cùng của thẻ thứ 2 với thuộc tính là. Nó sẽ được sử dụng mỗi khi ứng dụng có thể phát hiện ra được những thay đổi của giá trị bên trong biến name và nó sẽ gắn giá trị này trở thành nội dung HTML rồi đặt bên trong thẻ thứ 2 này.



Hình 2.14. Angular

* + 1. ***Ký số, mã mật file pdf sử dụng Docotic.Pdf library***

Docotic.Pdf là thư viện C # PDF hiệu suất cao cho .NET 4.0, .NET Standard 2.0 và các khuôn khổ mới hơn. Bạn có thể sử dụng nó để tạo, đọc và chỉnh sửa tài liệu, ký số, mã mật file PDF PDF trong các ứng dụng .NET Core, ASP.NET, Windows Forms, WPF, Xamarin, Blazor, Unity và HoloLense.

Docotic.Pdf cung cấp một API dễ sử dụng. Có một tập hợp lớn các mẫu C # và VB.NET để có thể tích hợp SDK vào dự án của mình một cách nhanh chóng.

Docotic.Pdf có thể tạo chữ ký điện tử bằng các thuật toán thông báo khác nhau (SHA-1, SHA-256, SHA-384, SHA-512, RIPEMD-160) và các định dạng chữ ký khác nhau (adbe.pkcs7.detached hoặc ETSI.CAdES.detached) . Thư viện cũng có thể xác minh chữ ký PDF.



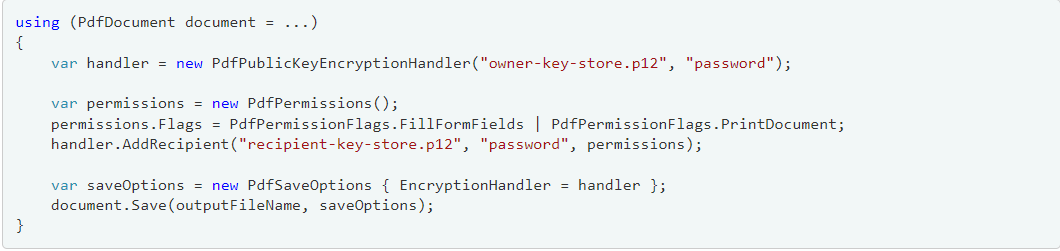
Hình 2.15. Ký số sử dụng thư viện Docotic.Pdf

Ngoài giúp thêm chữ ký số, Docotic.Pdf cũng hỗ trợ mã hóa tài liệu PDF sử dụng khóa công khai.

Với sự trợ giúp của thư viện Docotic.Pdf, bạn có thể mã hóa tài liệu PDF bằng mật khẩu hoặc chứng chỉ.

Thư viện mã hóa các tài liệu được bảo vệ bằng cách sử dụng một thuật toán mã hóa do bạn lựa chọn. Thư viện hỗ trợ các thuật toán mã hóa RC4 40-bit, RC4 128-bit, AES 128-bit và AES 256-bit.

Phần này chỉ đề cập đến mã hóa sử dụng chứng chỉ



Hình 2.16. Mã hóa file PDF sử dụng thư viện Docotic

Đầu tiên cần trích suất khóa công khai trong file .pfx bằng cách gọi hàm PdfPublicKeyEncryptionHandler() với tham số truyền vào là đường dẫn của file .pfx và mật khẩu xác thực. Tham số tiếp theo là permission là đại diện cho các quyền mà người dùng có thể có.

AddRecipient: Thêm người nhận mới với các quyền được chỉ định

Bên cạnh đó thư viện còn hỗ trợ các quyền khác như trong bảng dưới đây

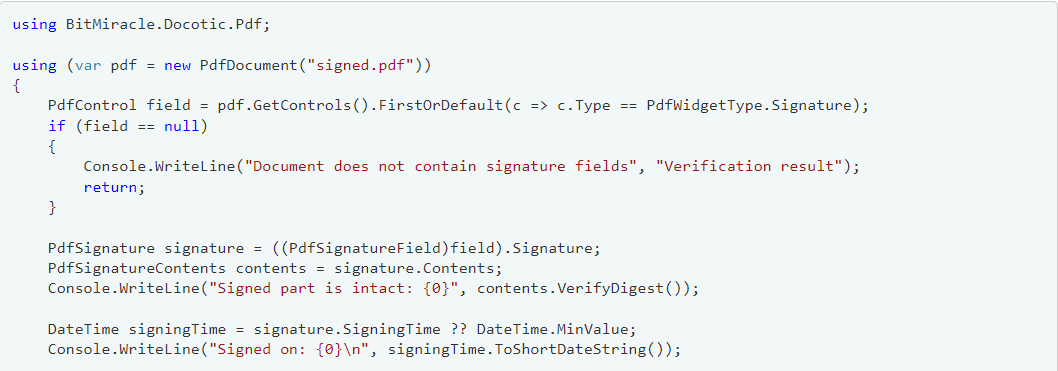
|  |  |
| --- | --- |
| PrintDocumment | Người dùng được phép in tài liệu. |
| AssembleDocument | Cho phép tập hợp tài liệu (chèn, xoay hoặc xóa các trang và tạo dấu trang hoặc hình ảnh thu nhỏ). |
| ExtractContents | được phép trích xuất văn bản và đồ họa (để hỗ trợ người dùng khuyết tật hoặc cho các mục đích khác). |
| ModifyContents | được phép sửa đổi nội dung của tài liệu |
| CopyContents | Người dùng được phép sao chép hoặc trích xuất văn bản và đồ họa từ tài liệu, kể cả việc sử dụng các công nghệ hỗ trợ như trình đọc màn hình hoặc các thiết bị hỗ trợ tiếp cận khác. |

Bảng 2.6. Ngoài ra còn hỗ trợ các thuật toán mã hóa khác nhau

|  |  |
| --- | --- |
| Tham số | Mô tả |
| PdfEncryptionAlgorithm.Aes128Bit | Thuật toán mã hóa AES sẽ được sử dụng với độ dài khóa là 128 bit. |
| PdfEncryptionAlgorithm.Aes256Bit | Thuật toán mã hóa AES sẽ được sử dụng với độ dài khóa là 256 bit. |
| PdfEncryptionAlgorithm.Standard128Bit | Thuật toán mã hóa RC4 sẽ được sử dụng với độ dài khóa là 128 bit. |
| PdfEncryptionAlgorithm.Standard40Bit | Thuật toán mã hóa RC4 sẽ được sử dụng với độ dài khóa là 40 bit. |

Bảng 2.7. Các tham số về thuât toán mã hóa

Thư viện Docotic.Pdf cung cấp các phương tiện để kiểm tra xem phần đã ký của tài liệu có bị thay đổi sau khi ký hay không. Bạn có thể kiểm tra xem chữ ký có chứa dữ liệu OCSP và / hoặc CRL được nhúng hay không. Và đối với bất kỳ chữ ký nào, có thể kiểm tra xem chứng chỉ ký của nó có bị thu hồi vào bất kỳ ngày nào đã định hay không.



Hình 2.17 Xác thực chữ ký số.

# **XÂY DỰNG ỨNG DỤNG WEB KÝ SỐ, MÃ MẬT FILE PDF**

* 1. **Tổng quan**

Hiện nay trên thế giới và tại Việt Nam đã có nhiều công cụ (phần mềm, giải pháp) hỗ trợ người dùng trong việc xác thực và mã hóa thông tin. Tuy nhiên các công cụ này chủ yếu được cung từ các nhà chữ ký số hoặc phần mềm miễn phí để cài đặt tại máy PC của người sử dụng. Việc tích hợp với các ứng dụng khác đặc biệt trong ứng dụng Web chưa được chú trọng.

Bài toán đặt ra là cần phải phát triển một hệ thống thông tin với giao diện người dùng trên nền web, có thể truy cập thông qua Internet. Người sử dụng của hệ thống này có thể gửi tài liệu đến máy chủ các tập tin có định dạng PDF. Website cung cấp các dịch vụ như ký số, mã hóa, giải mã dựa trên chứng thư số.

Để giải quyết yêu cầu bài toán trên, em đi xây dựng ứng dụng web ký số và mã mật file PDF dựa trên mô hình Client – Server, sử dụng chứng thư số ( định dạng PKCS#12) để thực hiện ký số , mã hóa và giải mã.

* 1. **Phân tích yêu cầu và thiết kế ứng dụng web**
     1. ***Phân tích yêu cầu***

Xây dựng một website ký số, hỗ trợ tạo chữ ký trên tệp PDF và chứng thực chữ ký số trên tệp PDF, …cho phép người dùng ký số mọi lúc mọi nơi

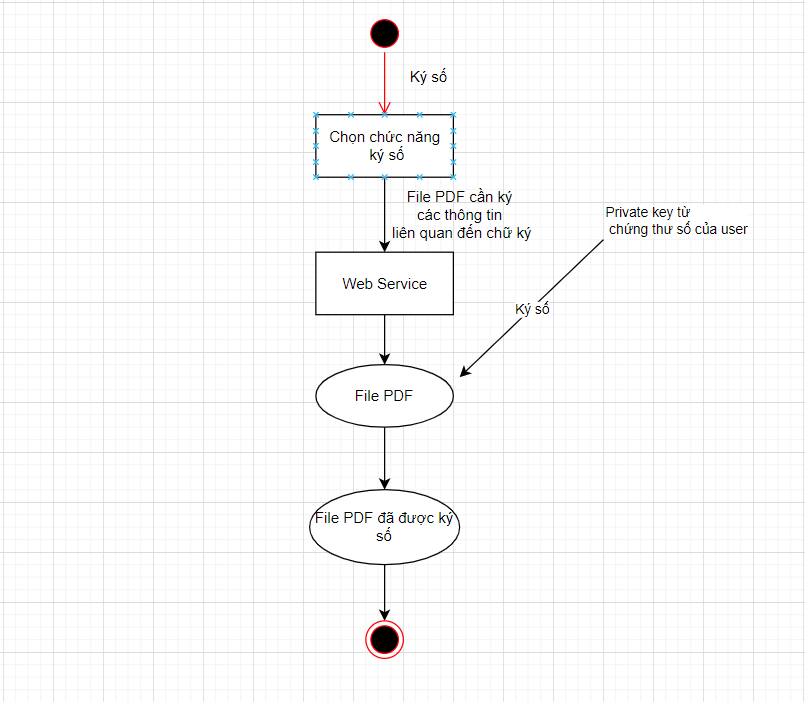
Gồm những yêu cầu sau:

* Ký số file PDF
* Xác thực chữ ký
* Mã hóa file PDF
* Giải mã file PDF
  + 1. ***Thiết kế ứng dụng web*** 
       1. Ký số

Dữ liệu đầu vào:

* Tệp PDF cần ký.
* Tệp .pfx của người ký, (PKCS#12 hoặc PFX) là định dạng nhị phân để lưu trữ chuỗi chứng chỉ và khóa riêng trong một tệp mã hóa duy nhất).
* Mật khẩu của tệp .pfx.

Dữ liệu đầu ra: Tài liệu PDF ban đầu chứa chữ ký của người ký.

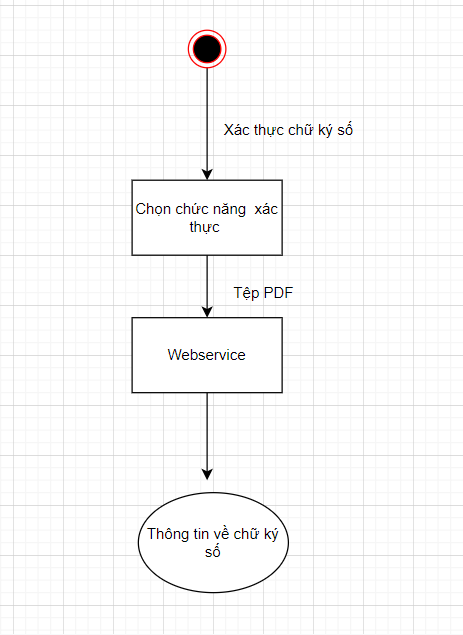


Hình 3.1. Sơ đồ chức năng ký số

* + - 1. Xác thực chữ ký số

Đầu vào: Tệp PDF

Đầu ra: thông tin về chữ ký số (đã ký hoặc chưa ký)



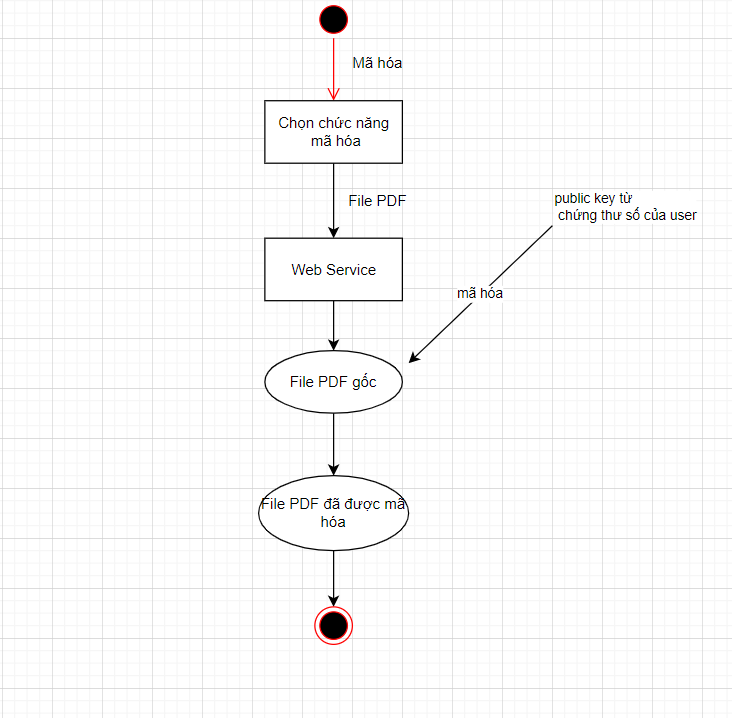
Hình 3.2. Sơ đồ xác thực ký số

* + - 1. Mã mật

Dữ liệu đầu vào:

* Tệp PDF cần mã hóa.
* Tệp pfx của người mã hóa
* Mật khẩu tệp pfx của người mã hóa
* Tệp pfx của người cần giải mã
* Mật khẩu tệp pfx của người cần giải mã

Dữ liệu đầu ra: Tệp PDF ban đầu đã được mã hóa. Tài liệu này đã được phân quyền khi người dùng thực hiện mã hóa.



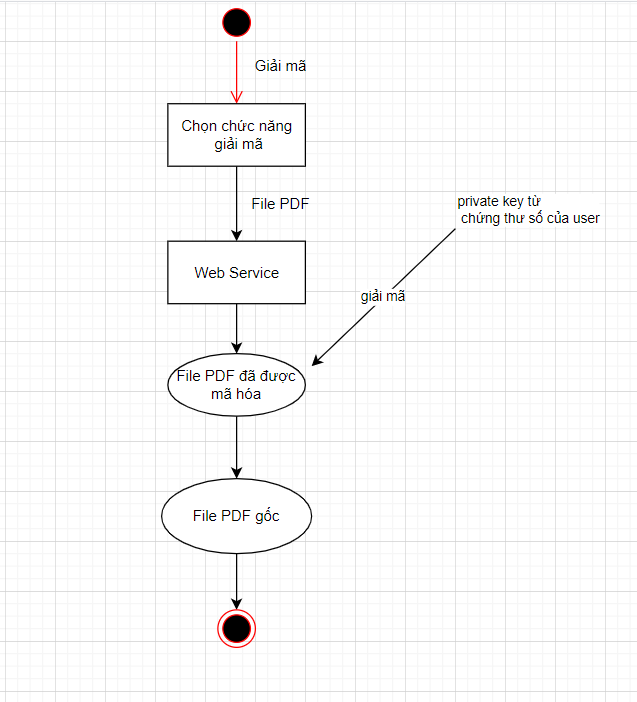
Hình 3.3. Sơ đồ chức năng mã hóa

* + - 1. Giải mã

Dữ liệu đầu vào:

* Tệp PDF đã được mã hóa
* Khóa công khai của người nhận (có thể là tệp .cer hoặc crt).

Dữ liệu đầu ra: Tệp PDF ban đầu đã được mã hóa. Tài liệu này đã được phân quyền khi người dùng thực hiện mã hóa.

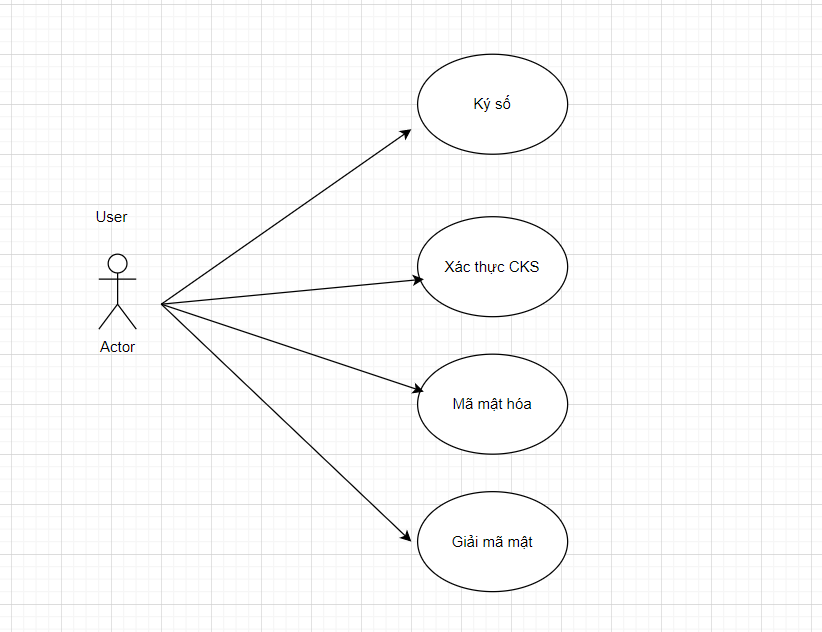


Hình 3.4. Sơ đồ chức năng giải mã

* 1. **Môi trường hệ thống**
     1. ***Các thành phần của hệ thống***

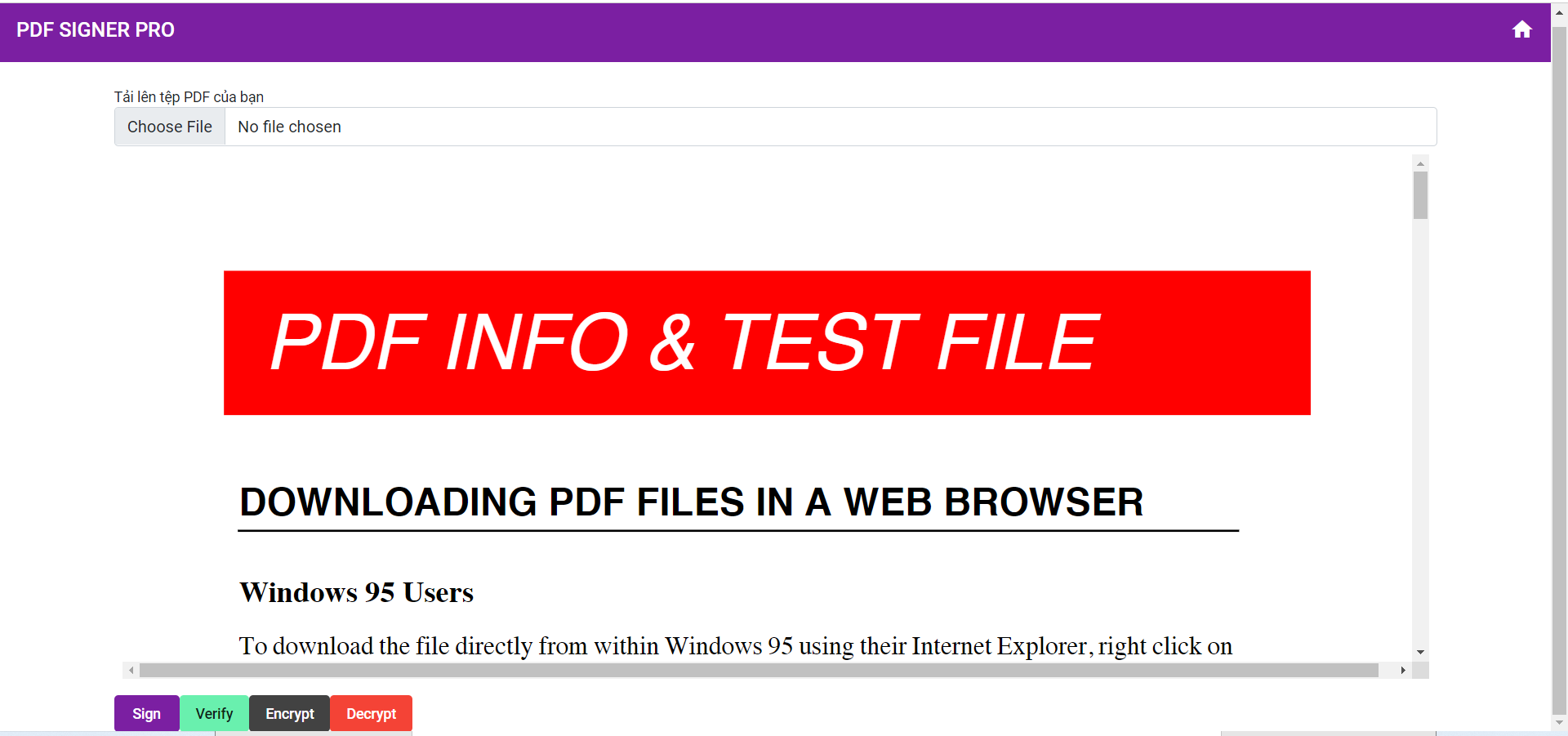
Hệ thống được thiết kế theo mô hình Client- Server. Vậy nên có 2 thành phần chính:

* Quản trị Server: Là người được quản lý server, khởi động, tắt Server.
* Người dùng: Mỗi người dùng là một client, tham gia sử dụng các chức năng
  + 1. ***Các chức năng của hệ thống***

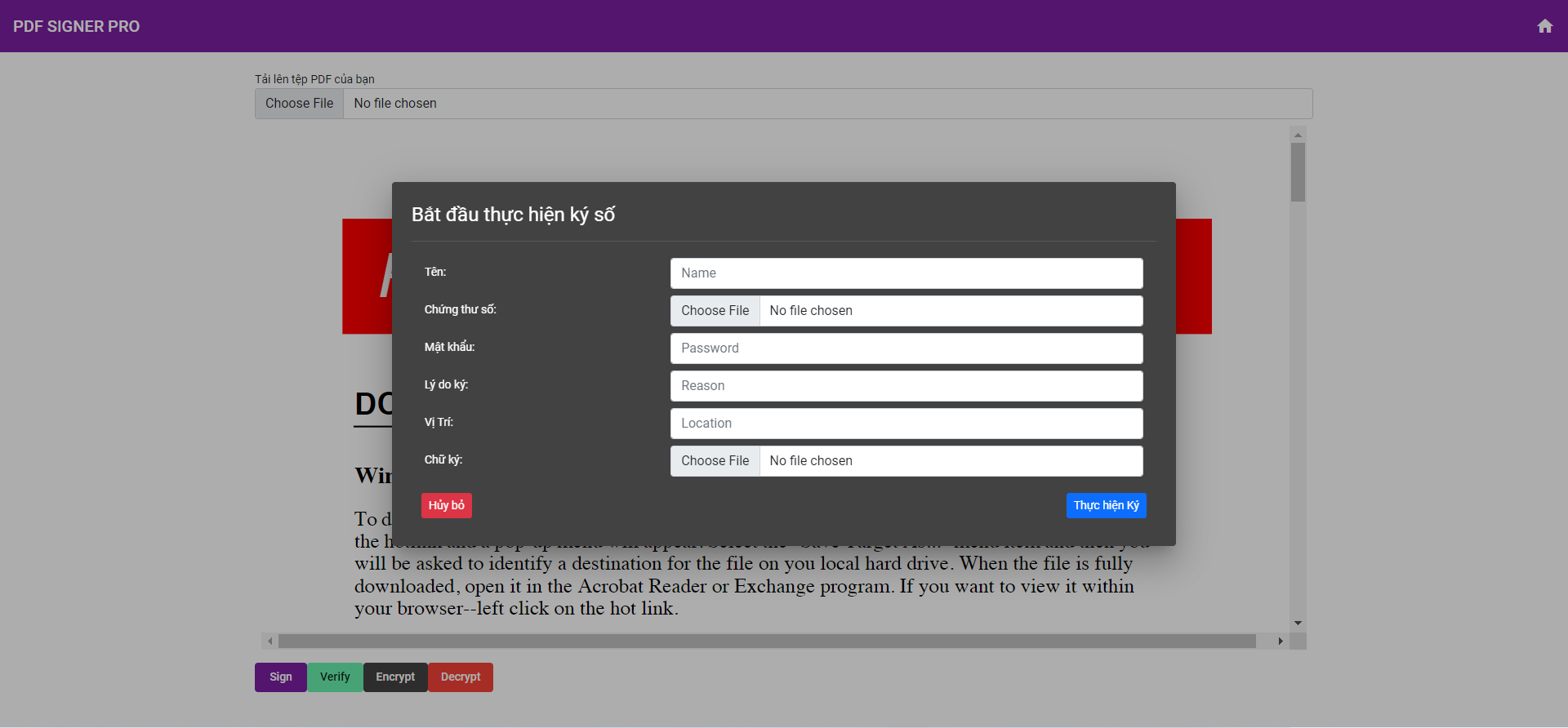


Hình 3.5. Chức năng chính của user

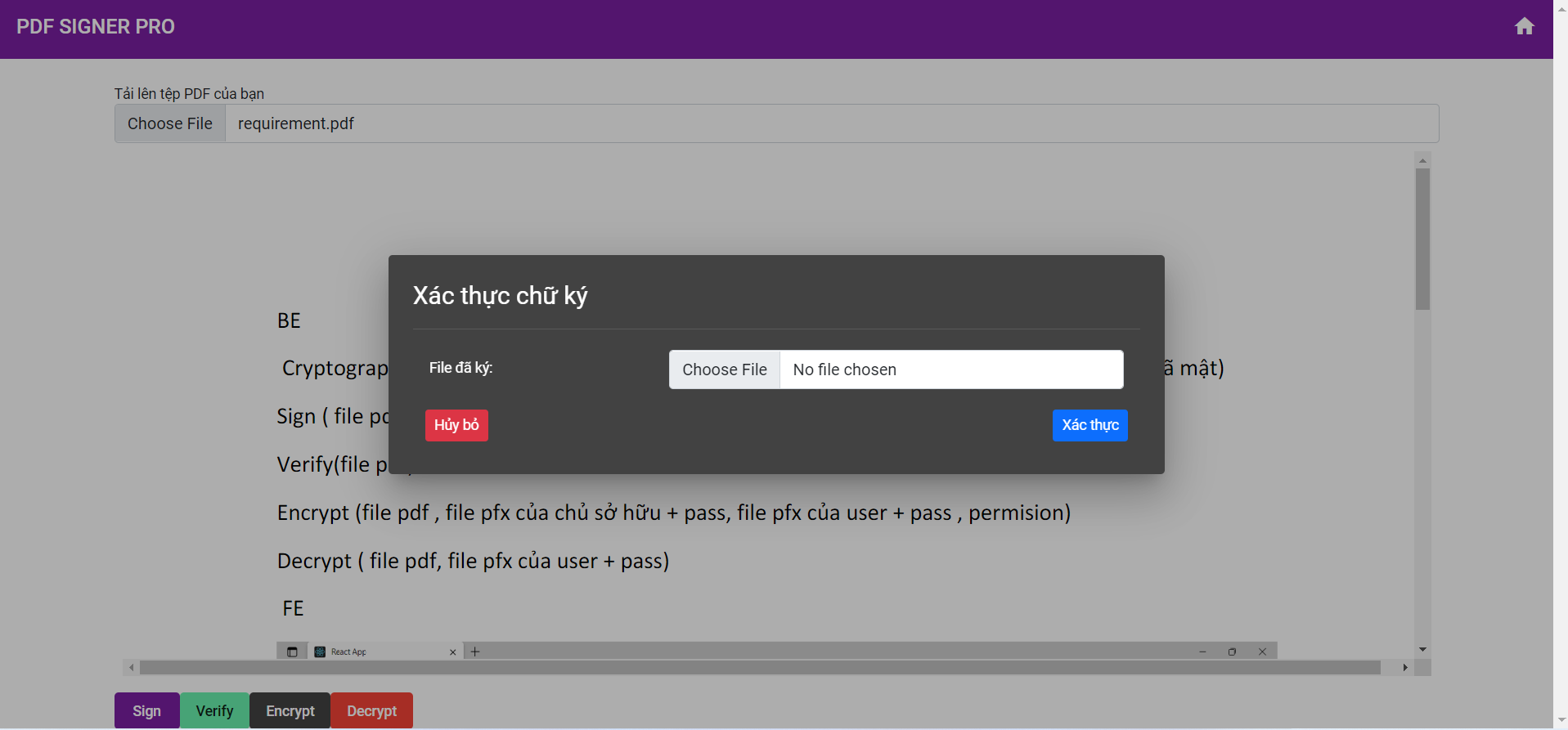
* 1. **Xây dựng website và thử nghiệm**
     1. ***Xây dựng website***



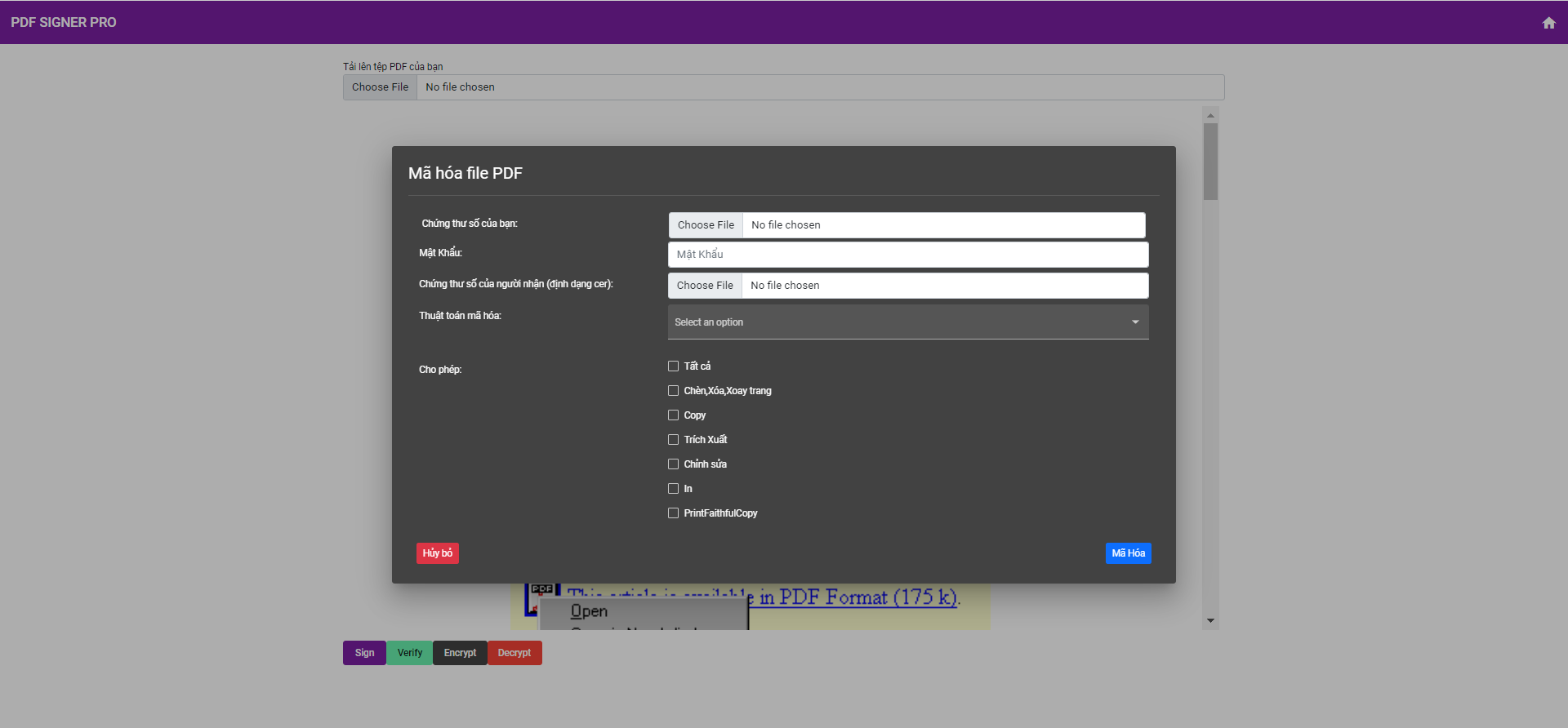
Hình 3.6. Giao diện trang chủ



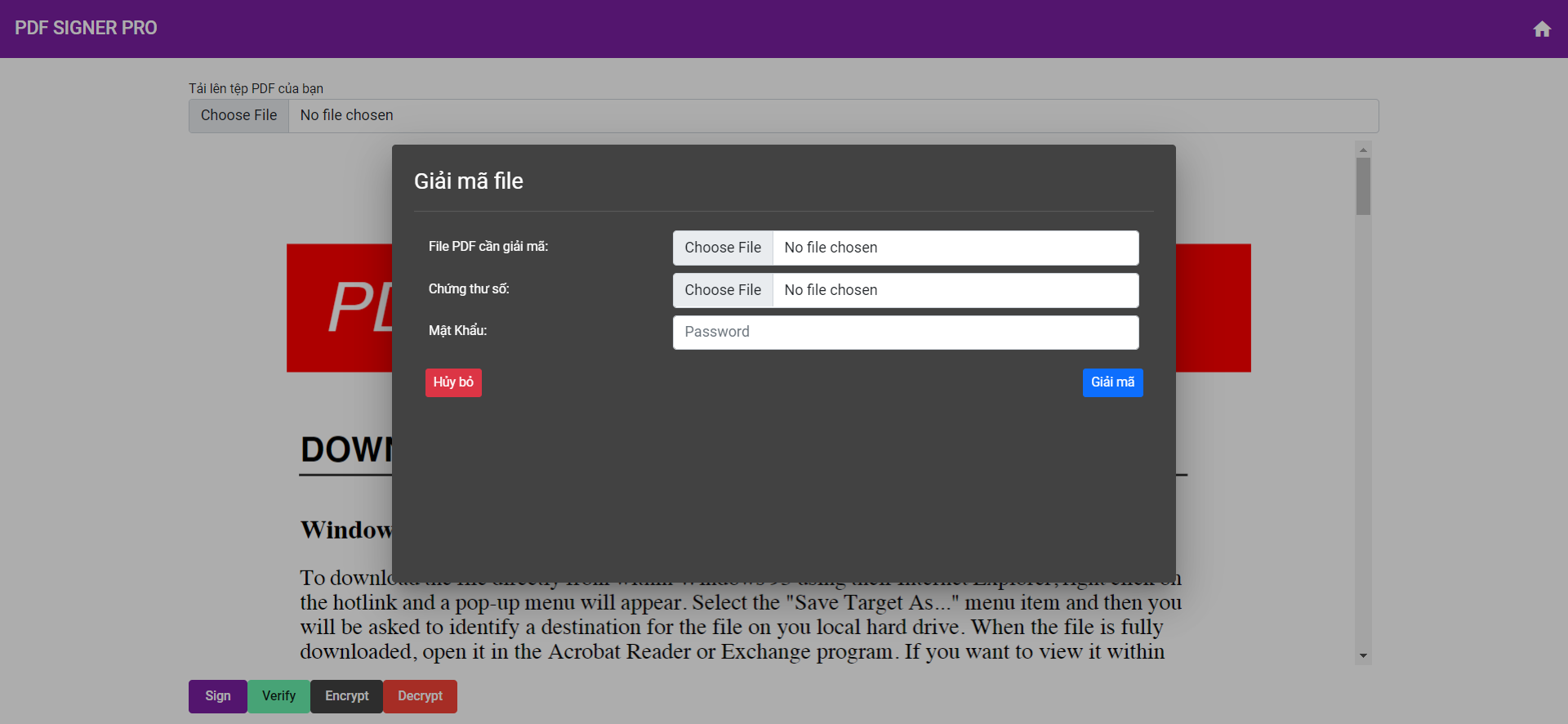
Hình 3.7. Giao diện chức năng ký số



Hình 3.8. Giao diện chức năng xác thực chữ ký

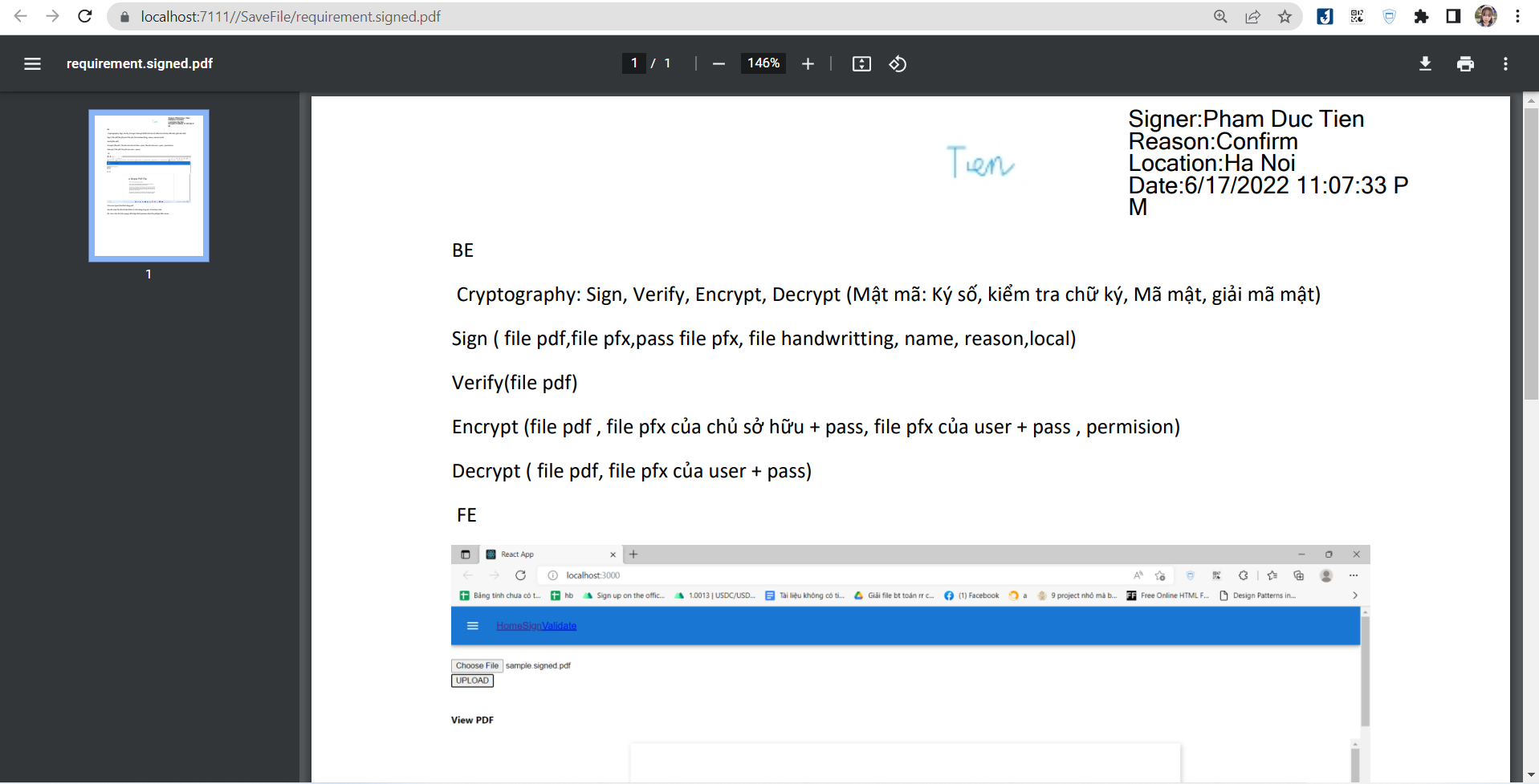


Hình 3.9. Giao diện chức năng mã hóa

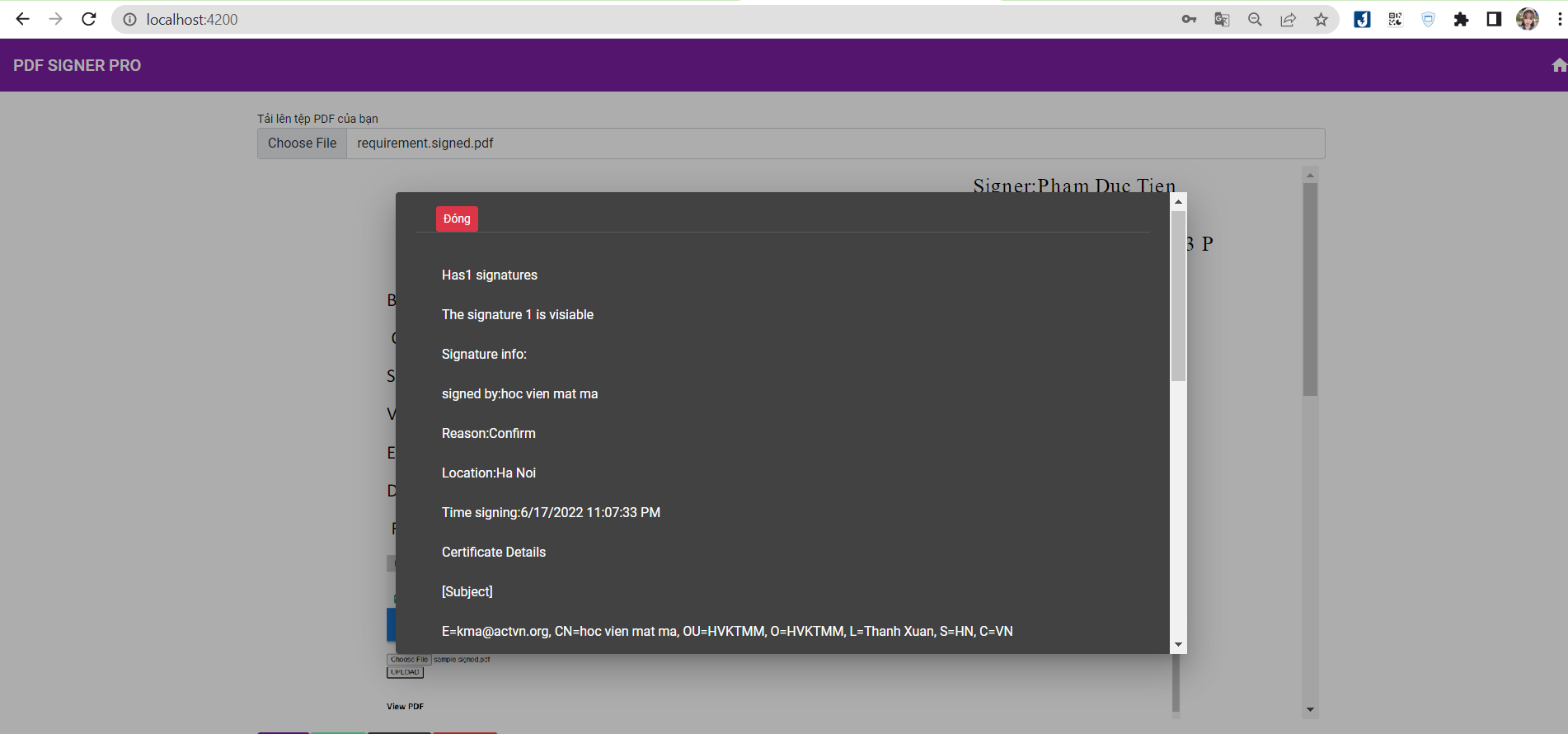


Hình 3.10. Giao diện chức năng giải mã

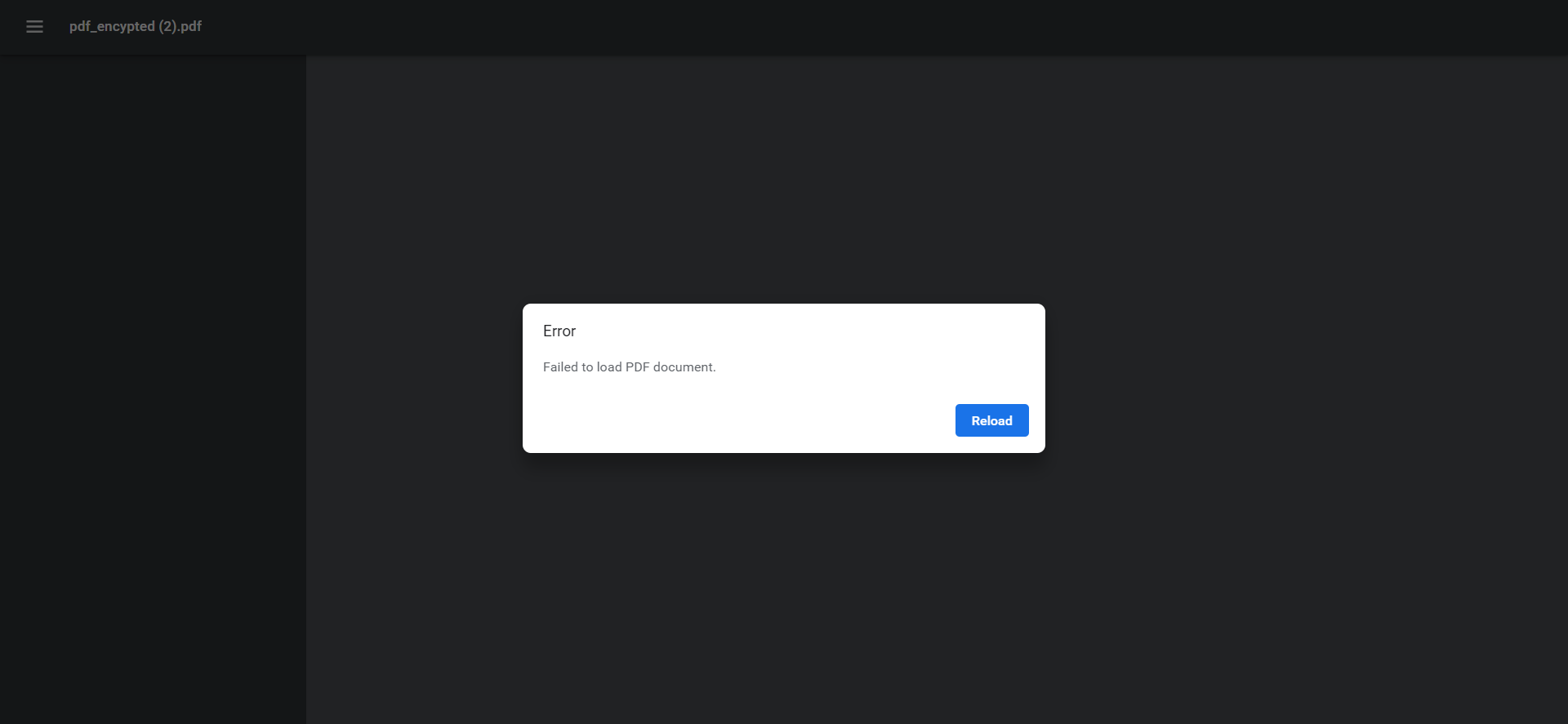
* + 1. ***Thử nghiệm***



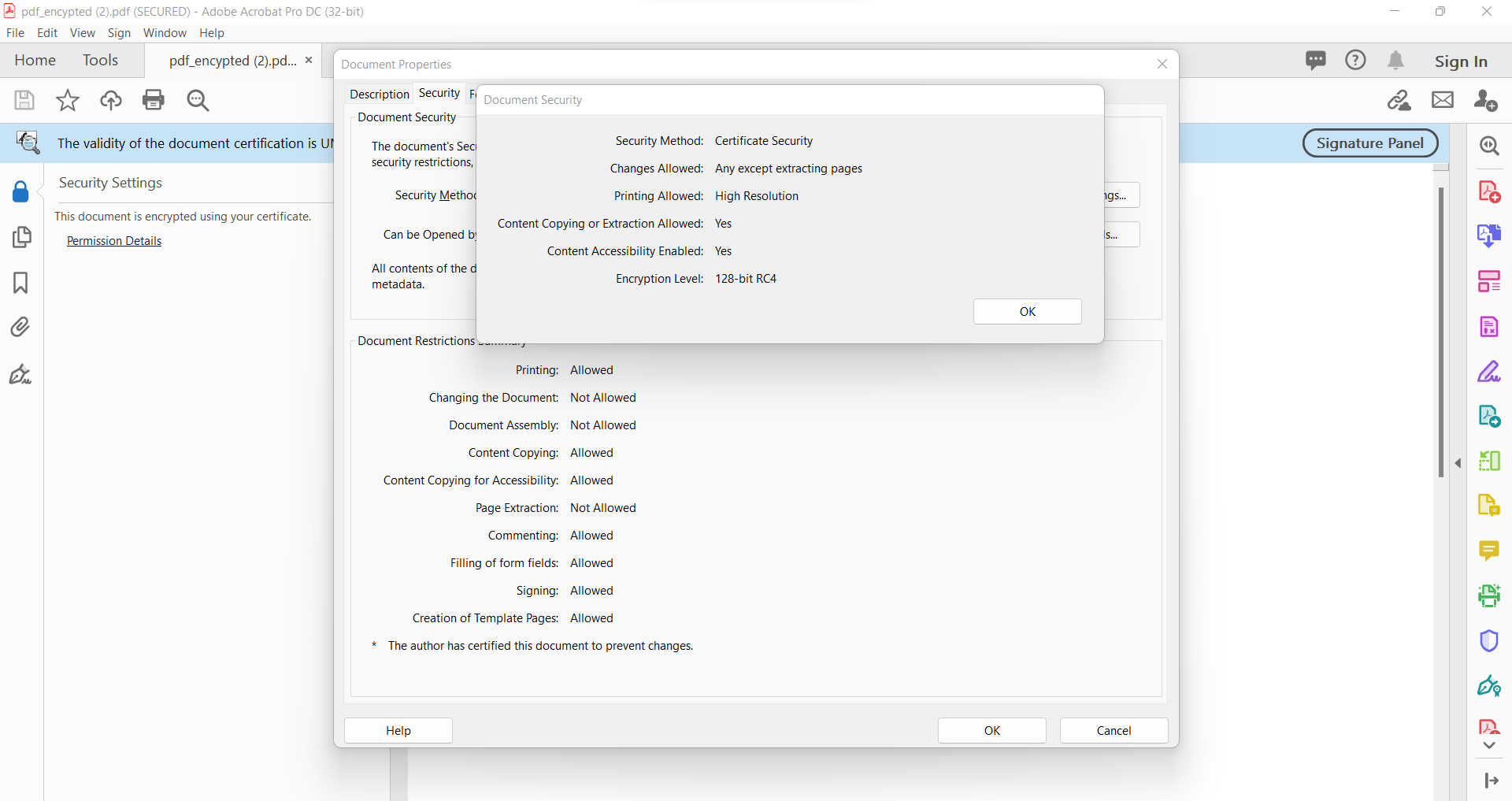
Hình 3.11. File sau khi ký số



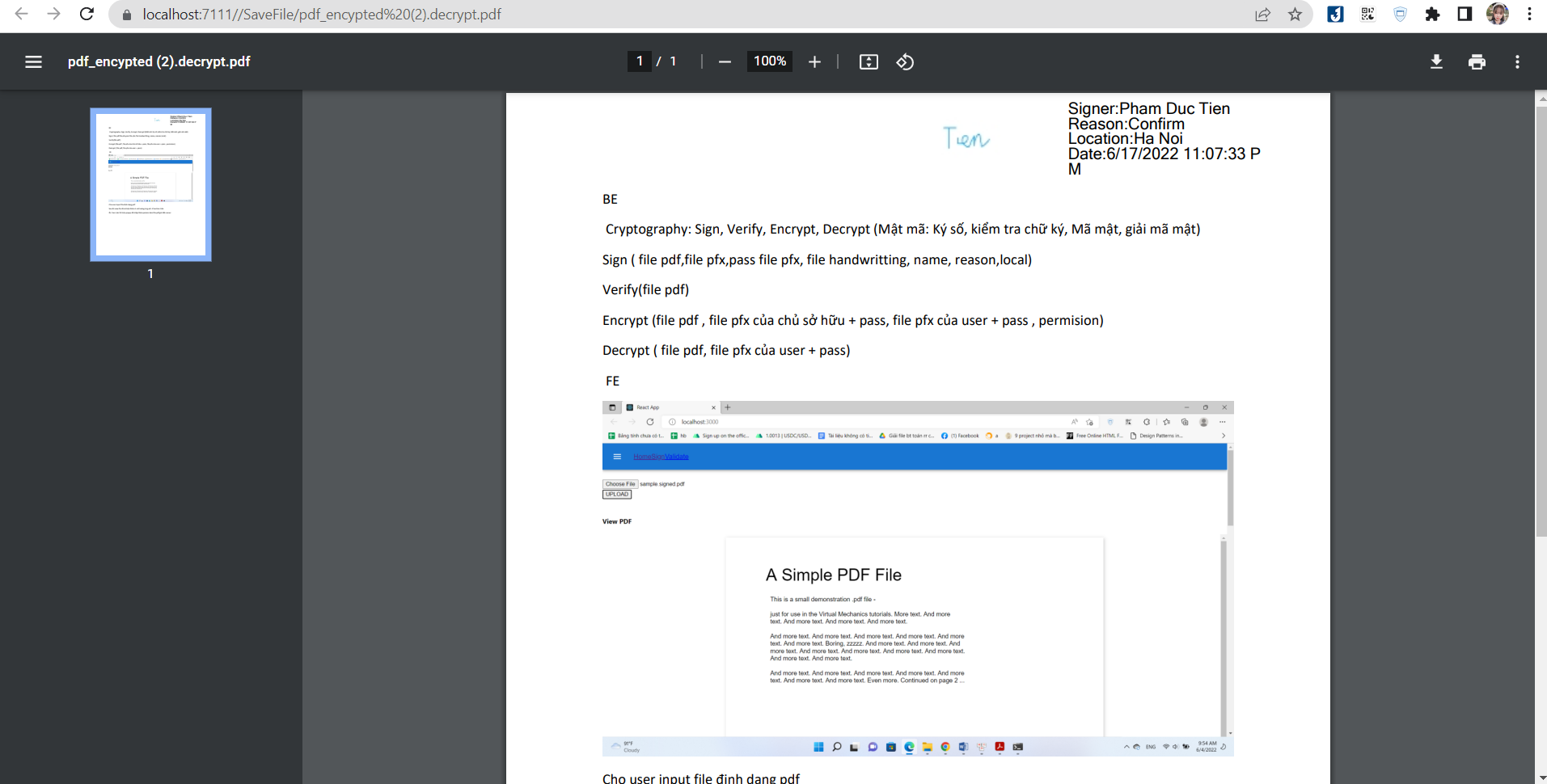
Hình 3.12. Xác thực chữ ký số



Hình 3.13. File đã được mã hóa và mở bằng trình duyệt thông thường



Hình 3.14. File đã mã hóa và mở bằng phần mềm Adobe Acrobat



Hình 3.15. File đã được giải mã

# **KẾT LUẬN**

**Kết quả đạt được:**

Trong thời gian tìm hiểu, xây dựng ứng dụng web, luận văn đã hoàn thành được các nhiệm vụ đặt ra, cụ thể là:

Về mặt lý thuyết: Luận văn nghiên cứu tìm hiểu hệ thống chứng thực điện tử, gồm:

* Cơ sở lý thuyết về mật mã khóa bí mật, mật mã khóa công khai, chữ ký số và hàm băm làm cơ sở cho việc tìm hiểu hạ tầng khóa công khai PKI.
* Hạ tầng khóa công khai PKI tìm hiểu về khái niệm PKI, các thành phần cũng như cách thức hoạt động, chức năng của PKI, các mô hình kiến trúc PKI.
* Các ứng dụng của PKI.
* Luận văn đi sâu vào nghiên cứu tìm hiểu về ứng dụng web ký số mã mật với định dạng tệp tin PDF trong PKI để giải quyết bài toán ký số mã mật mọi lúc mọi nơi

Về ứng dụng: Kết quả xây dựng ứng dụng ký số , mã mật file PDF trên nền tảng web. Tuy nhiên vẫn chưa hoàn thiện và tối ưu.

**Hướng phát triển:**

Tiếp tục tìm hiểu, nghiên cứu và phát triển ứng dụng web cho hoàn thiện hơn, thân thiện với người dùng và khắc phục những điểm yếu để đảm bảo tính bảo mật.

# **DANH MỤC TÀI LIỆU THAM KHẢO**

[1] Lê Quang Tùng, Nguyễn Thị Hồng Hà , giáo trình Chứng thực điện tử , Học Viện Kỹ Thuật Mật Mã (2013)

[2] Security of PDF Signatures

[3] PDF 2.0, ISO 32000-2 (2017, 2020)

[4] PDF32000\_2008

[4]  IETF Public-Key Infrastructure X.509 (PKIX) Working Group

[5] <https://bitmiracle.com/pdf-library/>

[6] Hồ Văn Hương, Hoàng Chiến Thắng, Ký số và xác thực trên nền tảng Web, tạp chí An toàn thông tin, 2013.