# Giải thuật mã hóa công khai RSA

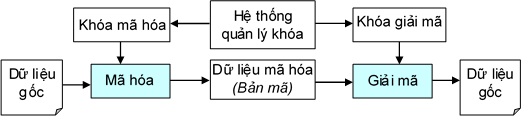
## RSA là gì?

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

### Một số khái niệm

Mã hóa dữ liệu là sử dụng một phương pháp biến đổi dữ liệu từ dạng bình thường sang một dạng khác, mà một người không có thẩm quyền, không có phương tiện giải mã thì không thể đọc hiểu được.

Giải mã dữ liệu là quá trình ngược lại, là việc sử dụng một phương pháp biến đổi dữ liệu đã được mã hóa về dạng thông tin ban đầu. Có thể mô tả quy trình thực hiện mã hóa dữ liệu và giải mã dữ liệu như sau:



***Hình 1-1: Quy trình mã hóa dữ liệu***

(Nguồn: *Handbook of Applied Cryptography*, A. Menezes, P . van Oorschot and S. V anstone)

Sau đây là một số khái niệm và kí hiệu liên quan về vấn đề mã hóa dữ liệu :

- E (Encryption - Mã hóa): Là quá trình chuyển đổi dữ liệu gốc thành dữ liệu được mã hóa sao người khác không thể đọc hiểu.

-D (Decryption - Giải mã): Là quá trình ngược lại của mã hóa, biến đổi dữ liệu đã được mã hóa thành dạng gốc ban đầu.

- M (Message - Thông điệp), bản gốc hay bản rõ (Plaintext): Là tệp dữ liệu chưa được mã hóa hoặc đã được giải mã.

- C (Ciphertext - Bản mã): Tệp dữ liệu đã được mã hóa.

- K (Key - Khóa): Là dãy các bít 0, 1 thường được đưa ra dạng xâu ký tự, số...

- KE: Là khóa dùng để mã hóa.

- KD: Là khóa dùng để giải mã.

Theo quy ước, khi mã hóa thì C = E(M) và khi giải mã thì M = D(C) = D(E(M)).

Theo phương pháp truyền thống, người ta thường dùng cùng một khóa để mã hóa và giải mã. Lúc đó khóa phải được giữ bí mật tuyệt đối. Người ta gọi đây là hệ thống mã hóa cổ điển (các tên gọi khác: đối xứng, một khóa, khóa bí mật...). Một phương pháp khác, sử dụng khóa công khai (còn gọi là phương pháp mã hóa bất đối xứng, hay hệ thống hai khóa) trong đó khóa để mã hóa và khóa để giải mã là khác nhau. Các khóa này tạo thành một cặp chuyển đổi ngược nhau và không khóa nào có thể suy ra được từ khóa kia. Quy trình mã hóa khóa công khai gồm có các bước cơ bản như sau:

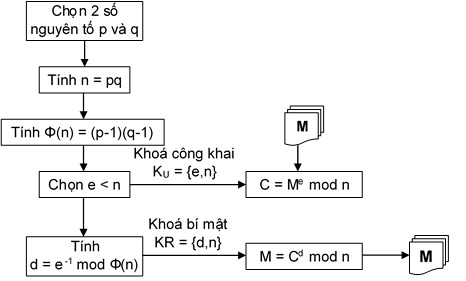
- Mỗi một hệ thống đầu cuối tạo một cặp khóa dùng cho quá trình mã hóa và giải mã mà hệ thống đó sẽ nhận.

- Mỗi hệ thống công bố khóa mã hóa của mình, gọi là khóa công khai, khóa còn lại gọi là khóa bí mật (khóa riêng) và phải được giữ an toàn.

- Nếu Bên gửi muốn gửi thông điệp cho Bên nhận, Bên gửi mã hóa thông điệp sử dụng khóa công khai của Bên nhận.

- Khi Bên nhận nhận thông điệp, Bên nhận giải mã bằng khóa riêng của Bên nhận.

Sơ đồ dưới đây minh họa các bước trong thuật toán RSA.



|  |  |
| --- | --- |
| Thuật toán RSA có hai Khóa:  **- *Khóa công khai*** (**Public key**): được công bố rộng rãi cho mọi người và được dùng để mã hóa  **-** ***Khóa bí mật*** (**Private key**):  Những thông tin được mã hóa bằng khóa công khai chỉ có thể được giải mã bằng khóa bí mật tương ứng | C:\Users\Mr Khiem\Desktop\RSA\Báo cáo\TQRSA.png |

## Các đặc điểm chính trong thuật toán mã hóa RSA

Thuật toán RSA được thiết kế dựa trên độ khó của bài toán phân tích ra thừa số nguyên tố trên tập số nguyên *Zn*.

Cho số nguyên dương n = p \* q, với p, q là 2 số nguyên tố rất lớn (ít nhất 100 ký số). Khi biết *n*, muốn tìm *p, q* thì phải giải bài toán phân tích ra thừa số nguyên tố, công việc này đòi hỏi phải thực hiện một số lượng các phép tính vô cùng lớn.

### a) Tạo khóa:

- Tạo ngẫu nhiên 2 số nguyên tố *p, q* khác nhau và rất lớn (có số ký tự ít nhất là 100), sau đó tính: *n = p \* q*; *Ф(n) = (p -1) \* (q -1)*.

- Chọn ngẫu nhiên 1 số *e* sao cho *1 < e < Ф(n)*, với *e* là số nguyên tố cùng nhau với *Ф(n)*.

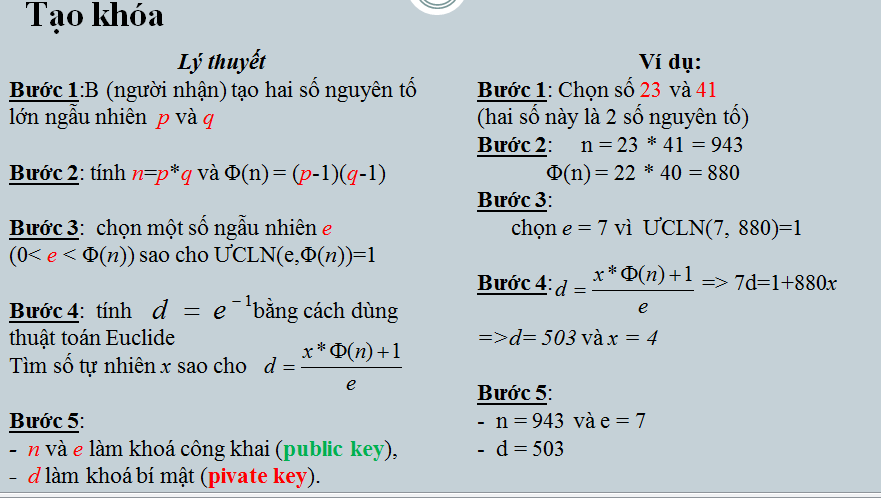
- Tính số nghịch đảo *d* của *e* đối với *Ф(n)*: *1 < d < Ф(n), ed = 1(mod Ф(n))*.

Ở đây *d* là số mũ bí mật; *e* là số mũ công khai.

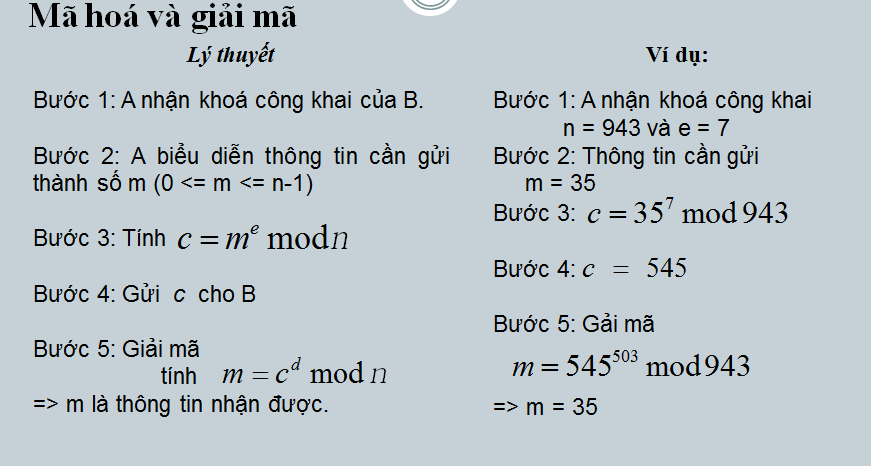
- Cặp *KU = {e, n}* được gọi là khoá công khai.

-Cặp *KR = {d, n}* được gọi là khoá bí mật.

### b) Mã hóa

Khi Bên gửi muốn gửi thông điệp *M* cho Bên nhận với yêu cầu cần bảo mật thông tin. Bên gửi yêu cầu Bên nhận gửi khoá công khai *KU = {e, n}*. Bên gửi dùng khoá công khai *KU* này để mã hoá thông điệp *M* thành *C* theo công thức: *C = Me  mod n*, sau đó Bên gửi gửi *C* cho Bên nhận.

c) Giải mã

Để giải mã bản mã *C*, Bên nhận dùng khoá bí mật *KR = {d, n}* để có thể khôi phục lại dữ liệu gốc ban đầu do Bên gửi gửi đến thông qua phép toán *M = Cd mod n*.

## Các vấn đề quan tâm của RSA

Độ an toàn của RSA đã được các nhà khoa học trên thế giới trình bày trong nhiều nghiên cứu, sau đây sẽ xem xét một số các phương pháp tấn công RSA:

- Vét cạn khóa: cách tấn công này thử tất cả các khóa *d* có thể có để tìm ra bản giải mã có ý nghĩa, tương tự như cách thử khóa *K* của mã hóa đối xứng. Với *n* lớn, việc tấn công là rất khó thực hiện.

- Phân tích *n* thành các thừa số nguyên tố: Hiện nay, có nhiều thuật toán phân tích mới đã được đề xuất, cùng với tốc độ xử lý của máy tính ngày càng nhanh, đã làm cho việc phân tích *n* không còn quá khó khăn như trước đây. Năm 1977, các tác giả của RSA đã treo giải thưởng để tấn công RSA có kích thước của *n* vào khoảng 428 bít, tức 129 chữ số. Các tác giả này ước đoán phải mất 40 nghìn triệu triệu năm mới có thể giải được. Tuy nhiên vào năm 1994, câu đố này đã được giải chỉ trong vòng 8 tháng. Một thiết bị lý thuyết có tên là TWIRL do Shamir và Tromer mô tả năm 2003 đã đặt ra câu hỏi về độ an toàn của khóa 1024 bít. Vì vậy, hiện nay người ta khuyến cáo sử dụng khóa có độ dài tối thiểu 2048 bít để bảo đảm an toàn. Để tránh gặp phải những vấn đề trên, RSA trên thực tế thường bao gồm một hình thức chuyển đổi ngẫu nhiên *M* trước khi mã hóa. Quá trình chuyển đổi này phải đảm bảo rằng *M* không rơi vào các giá trị không an toàn. Sau khi chuyển đổi, mỗi bản rõ khi mã hóa sẽ cho ra một trong số khả năng trong tập hợp bản mã. Điều này làm giảm tính khả thi của phương pháp tấn công lựa chọn bản rõ (một bản rõ sẽ có thể tương ứng với nhiều bản mã tuỳ thuộc vào cách chuyển đổi).

- Đo thời gian: Đây là một phương pháp phá mã không dựa vào mặt toán học của thuật toán RSA, mà dựa vào một “hiệu ứng lề” sinh ra bởi quá trình giải mã RSA. Hiệu ứng lề đó là thời gian thực hiện giải mã. Giả sử người phá mã có thể đo được thời gian giải mã *M = Cd mod n* dùng thuật toán bình phương liên tiếp. Trong thuật toán bình phương liên tiếp, nếu một bít của *d* là 1 thì xảy ra hai phép mô-đun, nếu bít đó là 0 thì chỉ có một phép mô-đun, do đó thời gian thực hiện giải mã là khác nhau. Bằng một số phép thử bản rõ chọn trước, người phá mã có thể biết được các bít của *d* là 0 hay 1 và từ đó biết được *d*. Phương pháp phá mã này là một ví dụ cho thấy việc thiết kế một hệ mã an toàn rất phức tạp. Người thiết kế phải lường trước được hết các tình huống có thể xảy ra.

## Ứng dụng

Tiêu chuẩn RSA được ứng dụng rộng rãi trong nhiều lĩnh vực như chữ ký số, thương mại điện tử, bảo mật, xác thực… Trong [Thông tư số 01/2011/TT-BTTTT](http://mic.gov.vn/vbqppl/Lists/Vn%20bn%20QPPL/DispForm.aspx?ID=7734) ngày 04/01/2011 của Bộ trưởng Bộ Thông tin và Truyền thông Công bố Danh mục tiêu chuẩn kỹ thuật về ứng dụng công nghệ thông tin trong cơ quan nhà nước quy định **Khuyến nghị áp dụng** tiêu chuẩn RSA, là một trong những giải thuật mã hóa và được xếp vào nhóm **Tiêu chuẩn về an toàn thông tin**.

1. **Chữ ký số (digital signature)**

**Digital Signature (**hay còn gọi là chữ ký điện tử) là một kĩ thuật xác thực cho phép người chủ nội dung của một thông điệp được quyền đính kèm một đoạn dữ liệu số như là chữ kí đánh dấu của người chủ đối với nội dung đã tạo ra. Về cơ bản, chữ kí số sẽ được tạo ra bằng cách hash nội dung thông điệp sau đấy mã hoá chuỗi hash bằng khoá bí mật (private key) của người chủ nội dung.

Digital signature là công cụ mật mã hóa (crytographic primitive) được sử dụng rất nhiều trong nền tảng công nghệ blockchain, chữ ký điện tử và hàm hash là nền tảng cơ bản mà công nghệ blockchain sử dụng.

**Chữ ký số**, digital signature, giúp ta đạt được những mục đích sau :

– **Non-repudiation** : có thể hiểu là giúp cho người nhận khi kiểm tra nội dung đã được ký chữ ký số kèm theo sẽ biết chắc người ký không thể chối cãi về những gì đã tạo ra ở thời điểm bắt đầu tạo chữ kí số.

– **Integrity** : chữ ký số giúp kiểm tra tính toàn vẹn dữ liệu của nội dung được gửi đi là không bị thay đổi hay chỉnh sửa kể từ lúc khởi tạo chữ ký số và ký vào văn bản gửi đi.

– **Authenticity** : chữ ký số cũng dùng để chứng thực nguồn gửi nội dung thông điệp đi. Thường thì thông tin về người chủ của chữ ký sẽ được thêm vào kèm với nội dung chữ ký số để giúp người nhận chứng thực được ai đã gửi thông điệp đi.

**Chữ ký số** ứng dụng nhiều trong các hoạt động cấp chứng chỉ Certificate SSL,… Chữ ký số cũng có có thể sử dụng trong các giao dịch thư điện tử, để mua bán hàng trực tuyến, đầu tư chứng khoán trực tuyến, chuyển tiền ngân hàng, thanh toán trực tuyến mà không sợ bị đánh cắp tiền như với các tài khoản Visa, Master.

## Vì sao chữ ký số lại quan trọng003F

Chữ ký số thường được sử dụng để đạt được tính toàn vẹn dữ liệu, xác thực và chống hoái thác.

* Tính toàn vẹn dữ liệu. Bob có thể xác minh rằng thông điệp của Alice đã không bị thay đổi trên đường được gửi đi. Bất kỳ sửa đổi trong thông điệp sẽ tạo ra một chữ ký hoàn toàn khác nhau.
* Tính xác thực. Miễn là khóa riêng của Alice được giữ bí mật, Bob có thể sử dụng khóa công khai của mình để xác nhận rằng chữ ký điện tử được tạo bởi Alice chứ không phải một ai khác.
* Chống thoái thác. Khi chữ ký đã được tạo, Alice sẽ không thể phủ nhận việc mình đã ký nó, trừ khi khóa riêng của cô ấy bị xâm phạm.

### ****I. Các yêu cầu thành phần để tạo ra được chữ ký số****

– Chữ ký số dựa trên công nghệ mã hoá khóa công khai (RSA): mỗi người dùng phải có 1 cặp khóa (key pair) bao gồm khóa công khai (public key) và khóa bí mật (private key).

+ **Private key** : là một khóa trong cặp khóa thuộc hệ thống mật mã không đối xứng, được dùng để tạo chữ ký số.

+ **Public key** : là một khóa trong cặp khóa thuộc hệ thống mật mã không đối xứng, được sử dụng để giả mã kiểm tra chữ ký số được tạo bởi khóa bí mật tương ứng trong cặp khóa.

+ **Digital Sign** (ký số) : là việc đưa khóa bí mật vào một chương trình phần mềm để tự động tạo và gắn chữ ký số vào thông điệp dữ liệu. Khi đọc các tài liệu chuyên ngành, bạn sẽ gặp nhiều cụm từ “**sign**” hoặc “**signed <object>**“, thì bạn phải hiểu là liên quan đến hoạt động của chữ kí số.

+ **Signer** (người ký) : là đối tượng dùng đúng khóa bí mật của mình để tạo chữ ký số và ký vào một thông điệp dữ liệu dưới tên của mình.

+ **Recipient** (người nhận) : là tổ chức, cá nhân nhận được thông điệp dữ liệu được ký số bởi người ký, sử dụng chứng thư số (digital certificate) của người ký đó để kiểm tra chữ ký số trong thông điệp dữ liệu nhận được và tiến hành các hoạt động, giao dịch có liên quan.

### ****II. Thế Digital Signature hoạt động như thế nào ?****

Hoạt động tạo ra chữ ký số Digital Signature yêu cầu phải có 1 cặp key **public & private**. Người kí sẽ giữ **private key** và dùng để **mã hoá thông tin tạo ra chữ kí số**, người nhận phải có **public key** của người kí tương ứng cặp với private key để **giải mã chữ kí số** nhằm đối chiếu thông tin.

Vậy ta có thể hình dung như thế này, bạn có 1 văn bản word hợp đồng và bạn cần gửi cho luật sư của bạn. Điều quan trọng là làm sao để luật sư biết chắc chắn văn bản word hợp đồng này được soạn thảo bởi chính bạn hay đã được xét duyệt bởi bạn.

**Quá trình kí (Signing)**

Lúc này **quá trình kí (Signing)** sẽ diễn ra như sau :

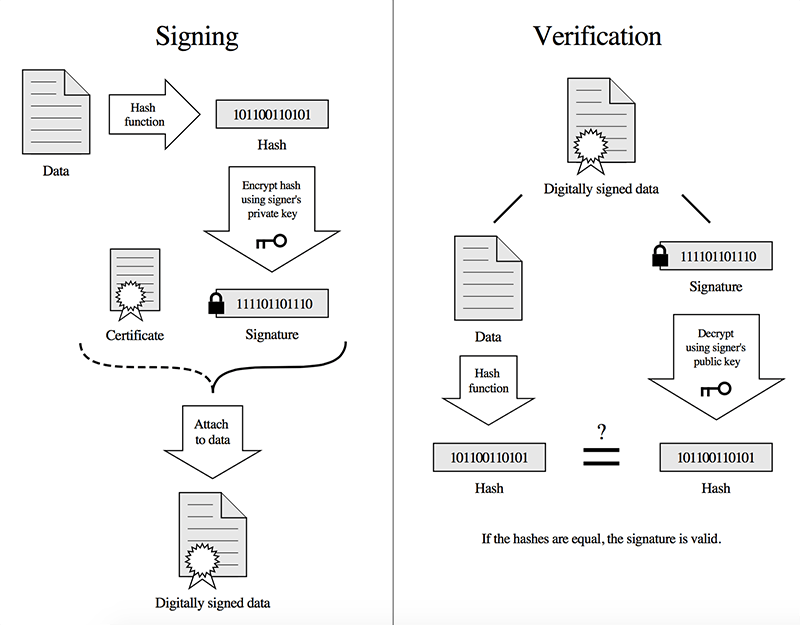
**1.** Ta có văn bản word hợp đồng .word (data).

**2.** Ta sử dụng chương trình**thuật toán băm** (MD5 hoặc SHA), băm văn bản word ra sẽ được 1 chuỗi kí tự duy nhất với độ dài cố định. Ta gọi là chuỗi H : (software) -> hash + word file -> string H .

– Thuật toán băm phải được thống nhất giữa 2 bên để có được kết quả chính xác giống nhau khi kiểm tra.

**3.** Sau đấy bạn sử dụng private key của bạn để mã hoá chuỗi băm từ văn bản word. Ta sẽ được 1 chuỗi ký tự hash đã được mã hoá gọi là cipher hash. Ta có thể gọi là chuỗi C : string H + private key người kí -> **cipher hash A**.

**4.** Lúc này “**cipher hash A”** chính là chữ kí số của người kí tạo ra là bạn (cipher hash A = digital signature). Nhiệm vụ lúc này của ta là thêm nó vào nội dung file word vào cuối văn bản, để bên kia nhận được nội dung văn bản và các chữ kí số.



## Vì sao nên sử dụng chữ ký số

Ngoài việc hỗ trợ các quy trình kinh doanh được thuận lợi cũng như ngăn chặn hành vi giả mạo các thông điệp và tài liệu quan trọng, việc sử dụng ký điện tử cung cấp thêm các lợi ích về xác nhận. Ngoài việc đảm bảo rằng tin nhắn hoặc tài liệu đi kèm không bị thay đổi trong quá trình chuyển tiếp, chữ ký điện tử cũng giúp ngăn chặn tình trạng bỏ sót những thay đổi chưa được xác nhận. Nếu nội dung được ký bị thay đổi, chữ ký điện tử sẽ bị vô hiệu, và gửi cảnh báo cho người gửi và người nhận về các vi phạm. Đó là do các hàm mã hóa được áp dụng sẽ ngăn không cho chữ ký mới và hợp lệ được tạo ra cho thông điệp đó.

Khi non-repudiation được kích hoạt, người gửi sẽ không thể phủ nhận việc đã ký thư điện tử vào ngày làm việc kế tiếp. Người nhận hoặc bất kỳ ai truy cập trái phép vào thư cũng sẽ bị ngăn chặn và không thể tạo chữ ký giả. Hầu hết các phương pháp non-repudiation cung cấp một dấu thời gian time-stamp không thể thay đổi và các bằng chứng về chữ ký số trong trường hợp private key đã bị xâm nhập hoặc bị thu hồi.

**Quá trình kiểm tra thông tin chữ kí số (Verification)**

Quá trình kiểm tra thông tin chữ ký số (Verification) sẽ diễn ra như sau :

**1.** Người nhận sẽ nhận được văn bản hợp đồng bao gồm nội dung chính + chữ kí số “**cipher hash A**” . Lúc này người nhận sẽ tách riêng nội dung văn bản hợp đồng và chữ kí số ra xử lý độc lập.

**2.** Ở phần nội dung văn bản hợp đồng, người nhận làm công việc giống người kí đó là ta sử dụng chương trình thuật toán băm (MD5 hoặc SHA) giống bên kí, băm văn bản word ra sẽ được 1 chuỗi kí tự duy nhất với độ dài cố định. Ta gọi là chuỗi H2 chẳng hạn sẽ đại diện cho chuỗi string hash của người nhận đã hash ra : (software) -> hash + word file -> **string H2**

**3.** Người nhận sử dụng public key của người kí (**signer’s public key**) để giải mã chuỗi “cipher hash A” đính kèm trong văn bản hợp đồng ra được chuỗi “**string H**” : cipher hash A + public key người kí -> **string H.**

**4.** Đối chiếu thông tin trùng khớp giữa string H và string H2 . Nếu khớp nhau tức nội dung chính xác, không bị thay đổi nội dung, chính xác của người đã tạo ra văn bản và hoàn tất quá trình kiểm tra thông tin chữ ký số (**verification**).

**Lưu ý :**

– Bất cứ thay đổi dù là nhỏ nhất vào nội dung thông điệp gửi đi (data) hay nôm na là văn bản hợp đồng sau khi đã khởi tạo chữ kí số cũng sẽ tạo ra kết quả hoàn toàn khác ở phía người nhận khi họ hash băm văn bản ra và đối chiếu với thông tin hash ở chữ kí số của bạn.

Chữ ký số cũng tồn tại một số vấn đề về bảo mật nhưng trong phạm vi bài viết này mình sẽ không đề cập đến. Kế đến chúng ta sẽ tìm hiểu nội dung liên quan sử dụng Chữ ký số nhiều nhất là “Digital Signature Certificate và Digital Certificate”.

## ****Mật mã hóa Khóa công khai (Public-key Cryptography)****

Digital Signature (chữ ký số) là chữ ký điện tử dùng trong blockchain được xây dựng trên hệ mật mã hóa khóa công khai hay còn được gọi là mật mã hóa khóa bất đối xứng (asymmetrical crytography).

Hệ thống này sử dụng một cặp khóa mật mã: khóa công khai như tên gọi của nó, được đưa ra cho mọi người biết, không yêu cầu bảo mật, và khóa cá nhân chỉ có chủ nhân mới biết, yêu cầu giữ bí mật. Để một người sử dụng hệ mật mã này, việc đầu tiên người đó sẽ dùng một thuật toán tạo khóa G để tạo ra một cặp khóa mật mã ký hiệu là (sk,pk).

Như ví dụ minh họa trên, thì sk sẽ là khóa bí mật của Alice, còn pk sẽ là khóa công khai của Alice.

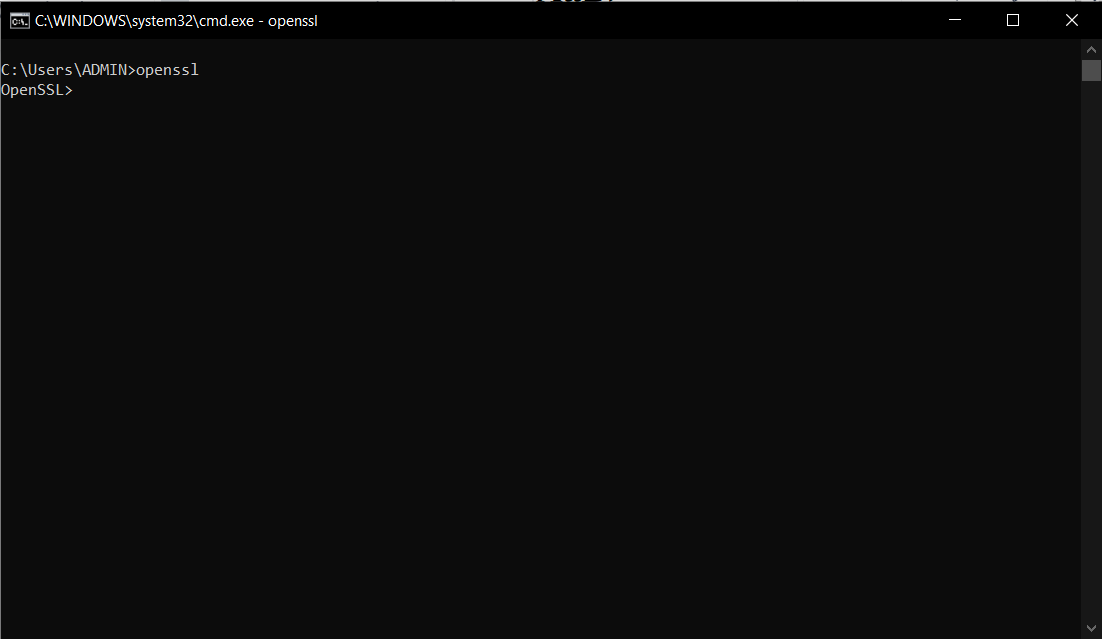
Bây giờ Bob có thể gửi một thông điệp m đến cho Alice bằng cách dùng khóa công khai của Alice pk để mã hóa c←E(m,pk), sao đó Bob gửi thông điệp đã được mã hóa c cho Alice. Alice muốn đọc được nội dung cả văn bản đã được mã hóa c thì sẽ dùng khóa bí mật (private key) của mình để giải mã m=D(c,sk). Người ta gọi đây là hệ mật mã hóa khóa bất đối xứng vì việc mã hóa E và giải mã D dùng hai khóa khác nhau.

Một hệ thống mã hóa dùng khóa công khai thì bao gồm 3 thuật toán chính (G,E,D): thuật toán sinh khóa G, thuật toán mã hóa E, và thuật toán giải mã D. Với hệ thống mã hóa này thì với mọi thông điệp m và mọi cặp khóa (sk,pk) được sinh ra bởi thuật toán G thì nếu ta dùng khóa bí mật sk để giải mã một nội dung được mã hóa bằng khóa công khai pk thì chúng ta sẽ phải có được nội dung gốc Dsk(Epk(m))=m.

## Tìm hiểu về công cụ sinh khóa và chữ ký số ( công cụ sử dụng OpenSSL)

#### Tạo khóa từ OpenSSl

Ở đây em dùng OpenSSL của window khi cài về gõ OpenSSL ta được giao diện như sau:



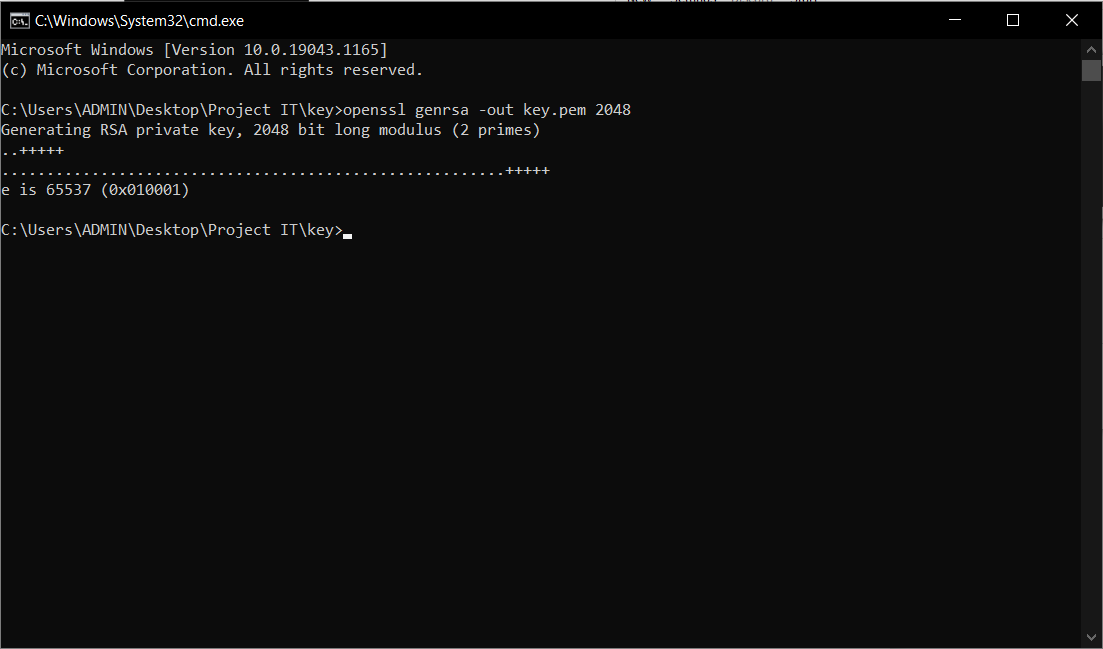
***Hình 2-1: Giao diện ở cmd khi hoàn tất việc cài đặt OpenSSL***

Sau đó ta bắt đầu tạo khóa:

B1: đầu tiên ta tạo file có tên là key.pem (file này là privatekey) bằng câu lệnh :

*openssl genrsa -out key.pem 2048*

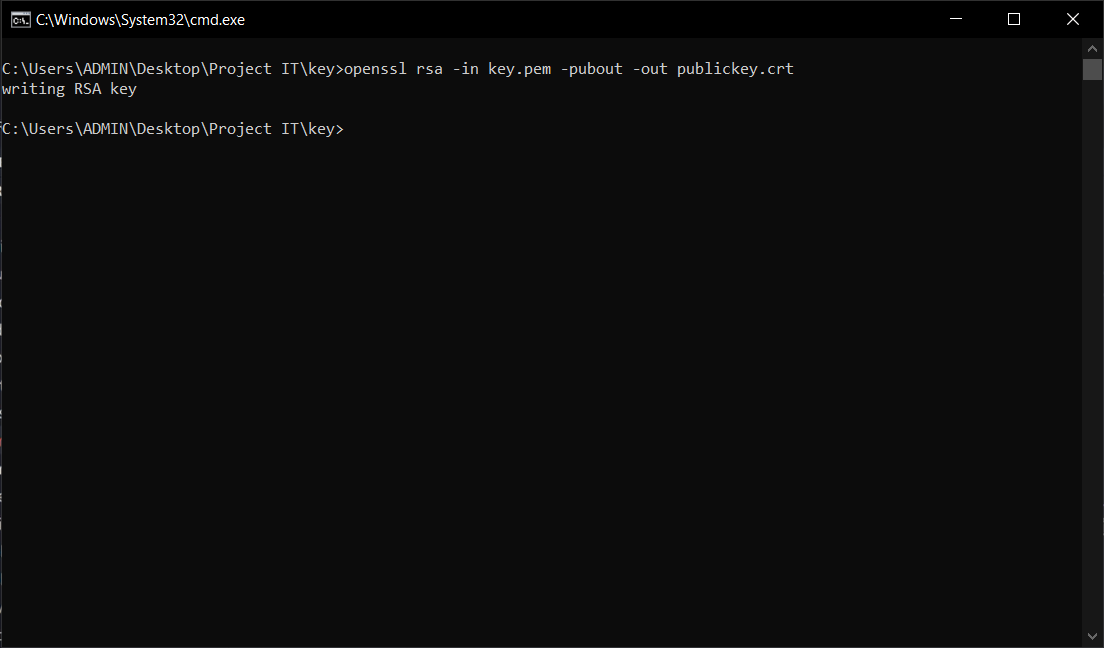
Với thông số cuối cùng là độ lớn của khóa được tín bằng bits (ở trên ta tạo file key.pem có độ lớn của khóa là 2048 bits)



***Hình 2-2: Ảnh tạo file private key***

B2: sau khi đã tạo xong file key.pem ta bắt đầu tạo public key bằng private key vừa tạo ở trên bằng câu lệnh:

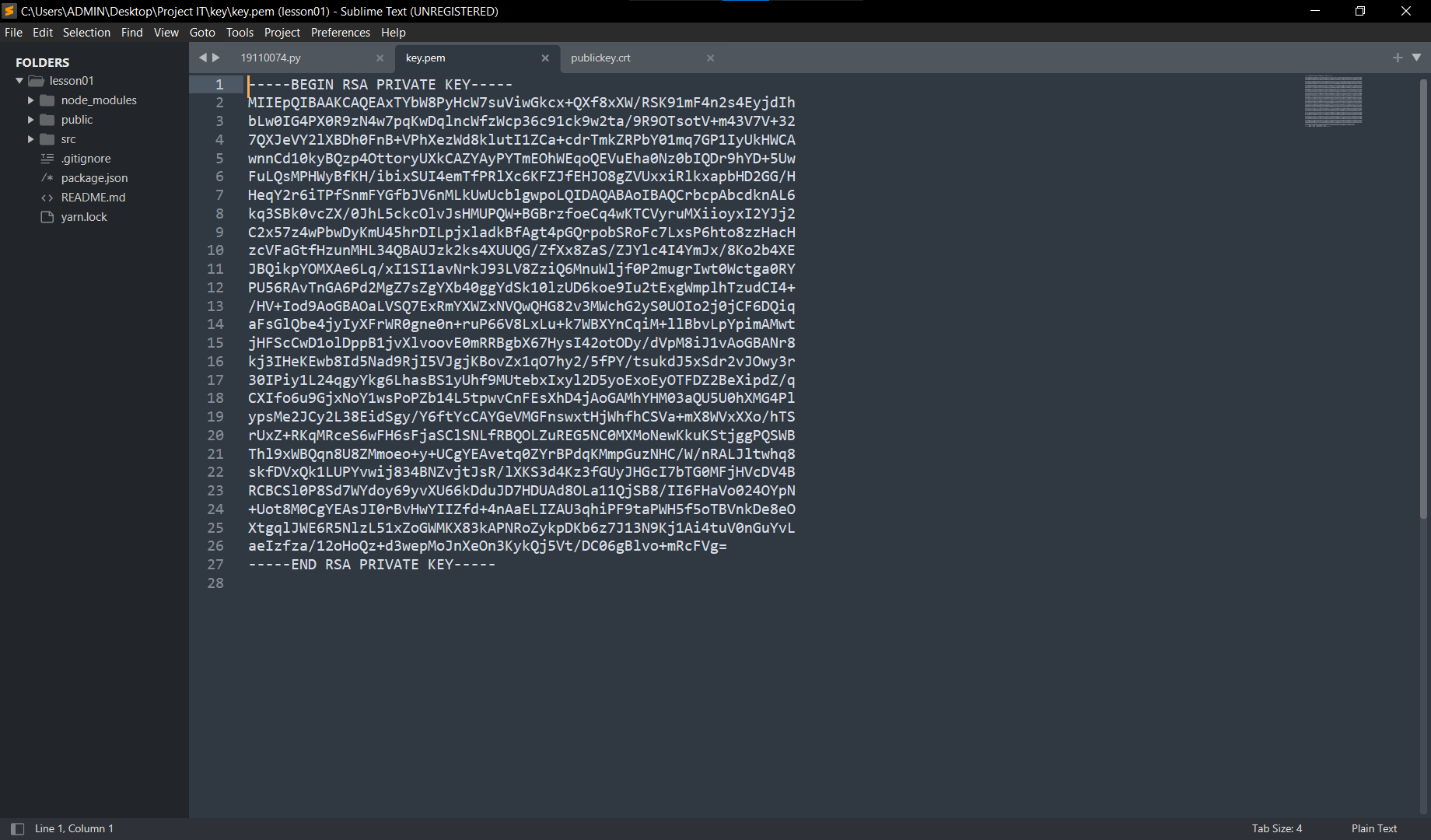
*openssl rsa -in key.pem -pubout -out publickey.crt*



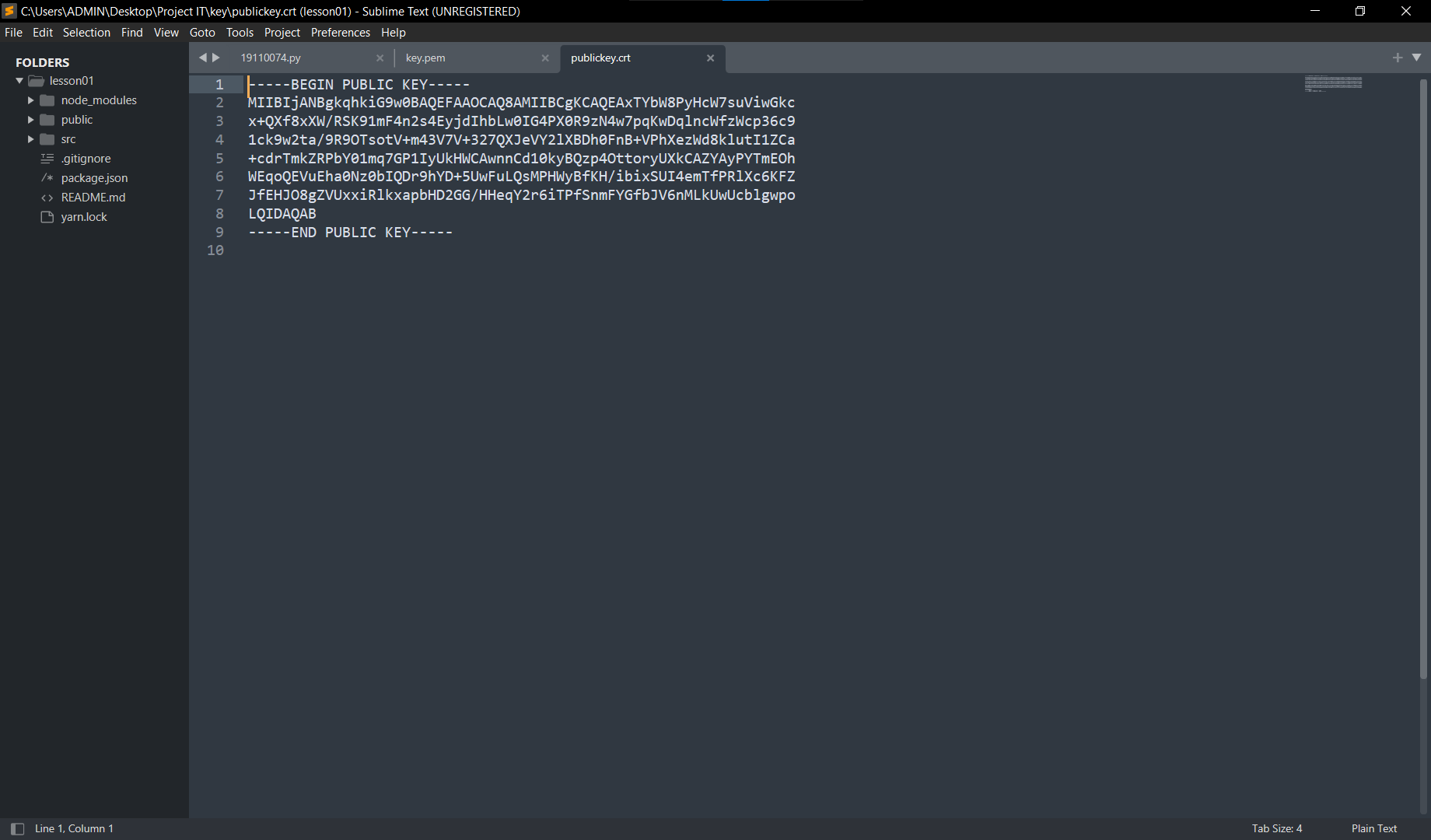
***Hình 2-3: Ảnh tạo file public key từ private key vừa tạo***

Sau khi tạo xong thì ta được public key và private key.

Mở 2 file public key và private key mà ta vừa tạo ta thấy được nội dung của key:



***Hình 2-4: Ảnh nội dung file private key***



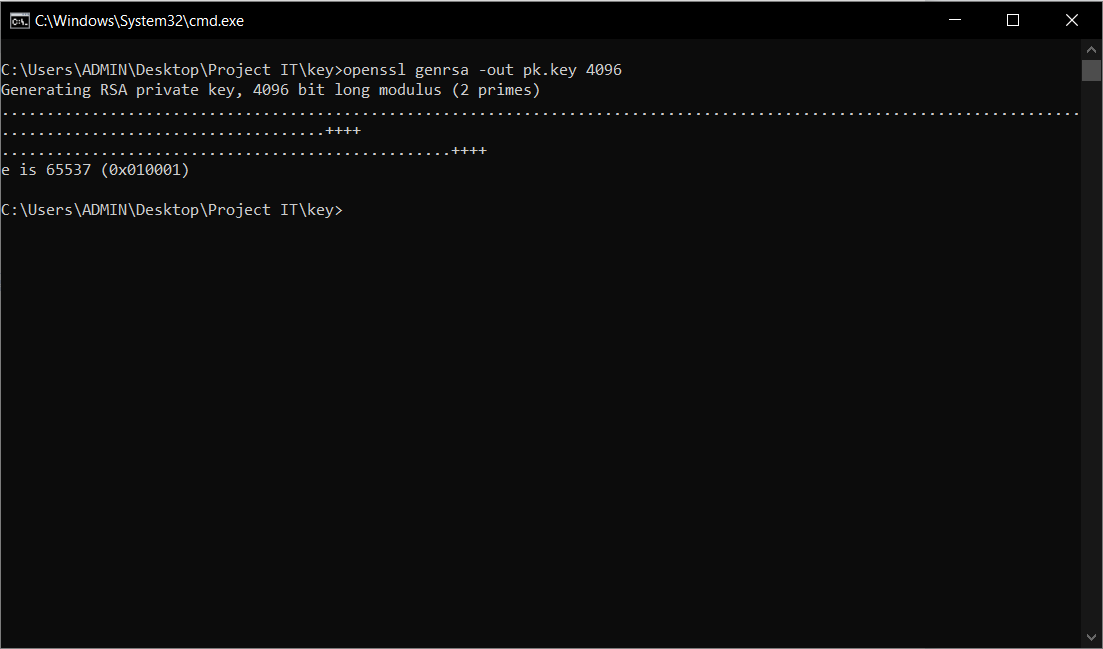
***Hình 2-5: Ảnh nội dung file public key***

#### Chữ ký số từ OpenSSL

B1: Cũng như tạo key thì bước đầu tiên ta tạo một file private bằng câu lệnh :

*openssl genrsa -out pk.key 4096*

Ở đây private key có độ lớn 4096 bits và được lưu vào tập tin pk.key



***Hình 2-5: Ảnh tạo private key pk.key***

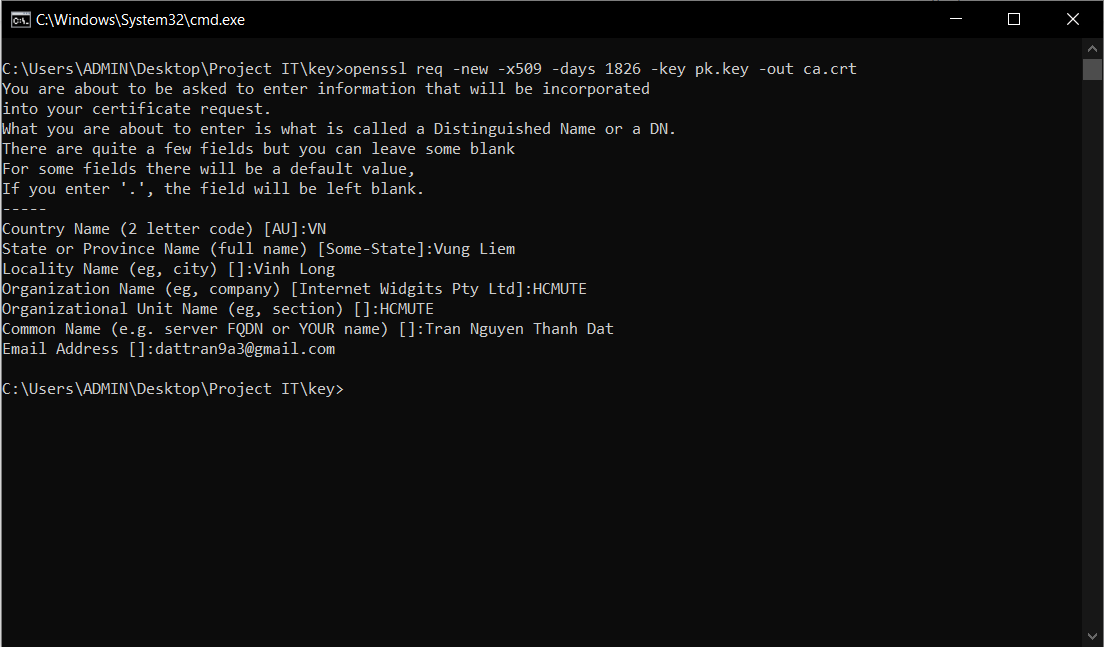
B2: Tạo một chứng chỉ từ khóa riêng (private key vừa tạo) bằng câu lệnh:

*openssl req -new -x509 -days 1826 -key pk.key -out ca.crt*

Với -days 1826 là ngày hết hạn của chứng chỉ , pk.key là khóa riêng vừa tạo, ca.crt là tập tin chứa chứng chỉ sau khi tạo.

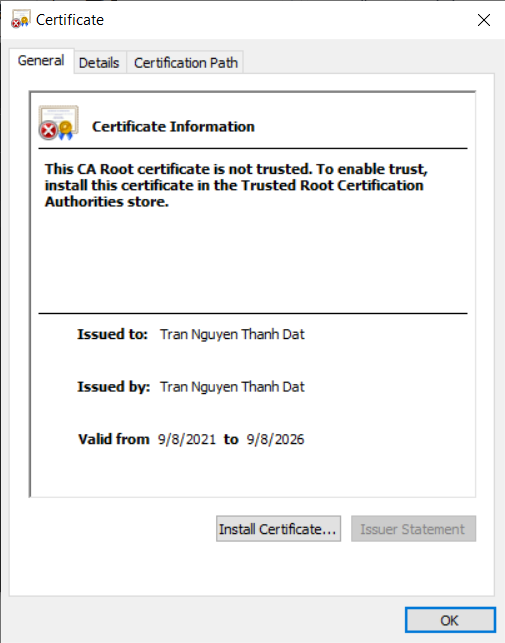
Tiến hành nhập vào một số thông tin cung cấp cho chứng chỉ này:

* Country Name: tên nước (chỉ nhập 2 kí tự,ví dụ Việt Nam thì nhập VN)
* State or Province Name: tên quận/huyện (ví dụ: Vung Liem)
* Locality Name: nơi sống (ví dụ : Vinh Long)
* Organization Name: tên công ty (ví dụ: HCMUTE)
* Organization Unit Name: tên đơn vị (ví dụ: HCMUTE)
* Common Name: tên của bạn (ví dụ: Tran Nguyen Thanh Dat)
* Email Address: địa chỉ email (ví dụ: dattran9a3@gmail.com)



***Hình 2-6: Ảnh khi tạo chứng chỉ ca.crt từ private key pk.key***

Sau khi chaỵ lệnh trên ta được ca.crt mở file lên ta được như hình:



***Hình 2-7: Ảnh nội dung chứng chỉ vừa tạo***

B3 : Tạo tập tin PFT

Từ private key (pk.key) và chứng chỉ (ca.crt) vừa tạo ta tạo tập tin PFX bằng câu lệnh :

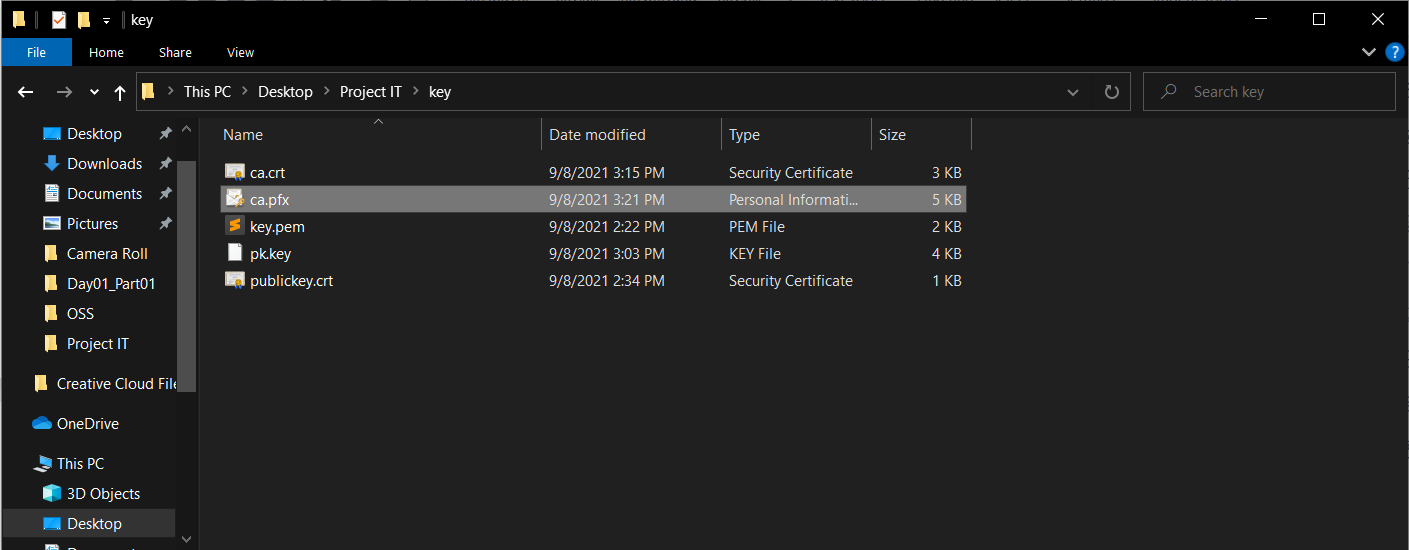
*openssl pkcs12 -export -out ca.pfx -inkey pk.key -in ca.crt*



***Hình 2-8: Ảnh tạo tập tin PFX***

Ở đây OpenSSL yêu cầu ta tạo mật khẩu để mở file ca.pfx ta tạo

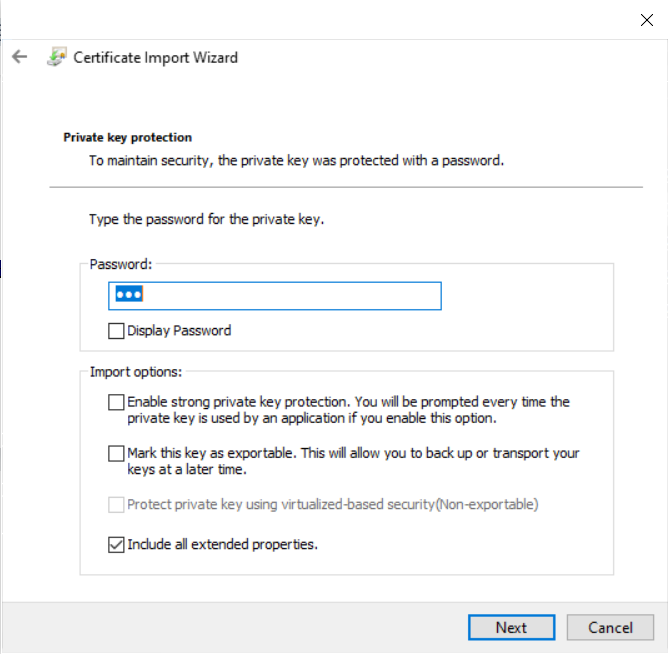
Kết quả ta nhận được tập tin ca.pfx



***Hình 2-9: Ảnh file ca.pfx vừa tạo***

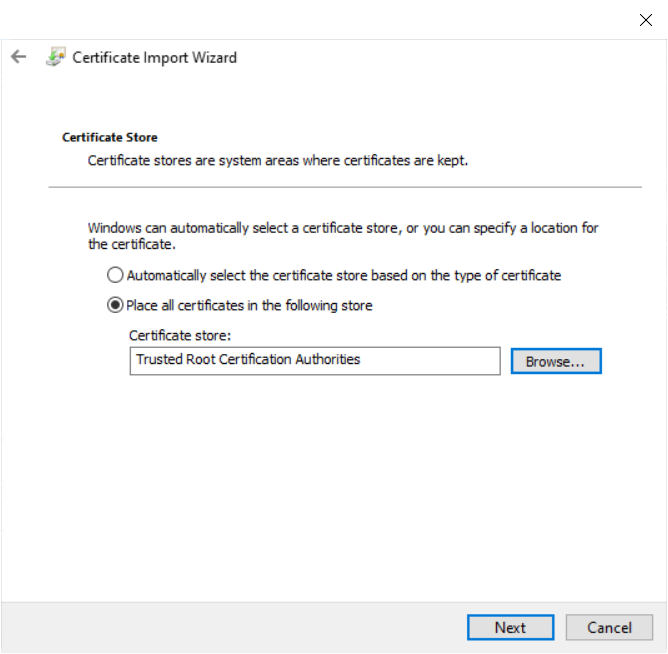
B4 : Cài đặt chữ ký điện tử

Ta chạy tập tin ca.pfx vừa tạo



***Hình 2-10: Ảnh chạy tập tin ca.pfx***

Đến đây nó yêu cầu ta cung cấp mật khẩu khi nảy tạo file pfx



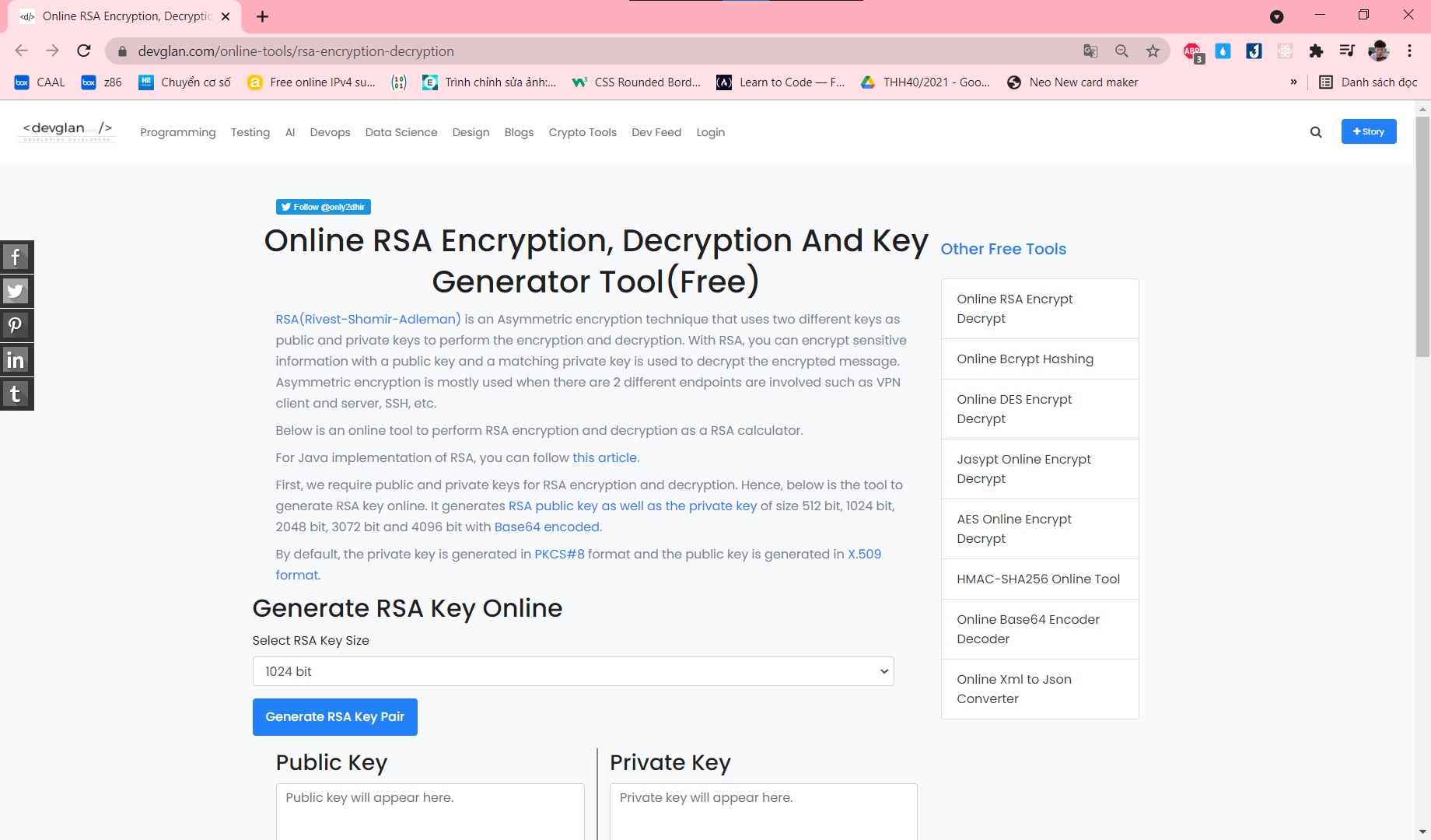
***Hình 2-11: Ảnh chọn nơi đăng ký chữ ký điện tử***

Tiếp đến ta chọn nơi đăng ký chữ ký điện tử là Trusted Root Certification Authorties

Sau khi hoàn tất thì ta ấn Finish để hoàn thành quá trình cài đặt chữ kí.

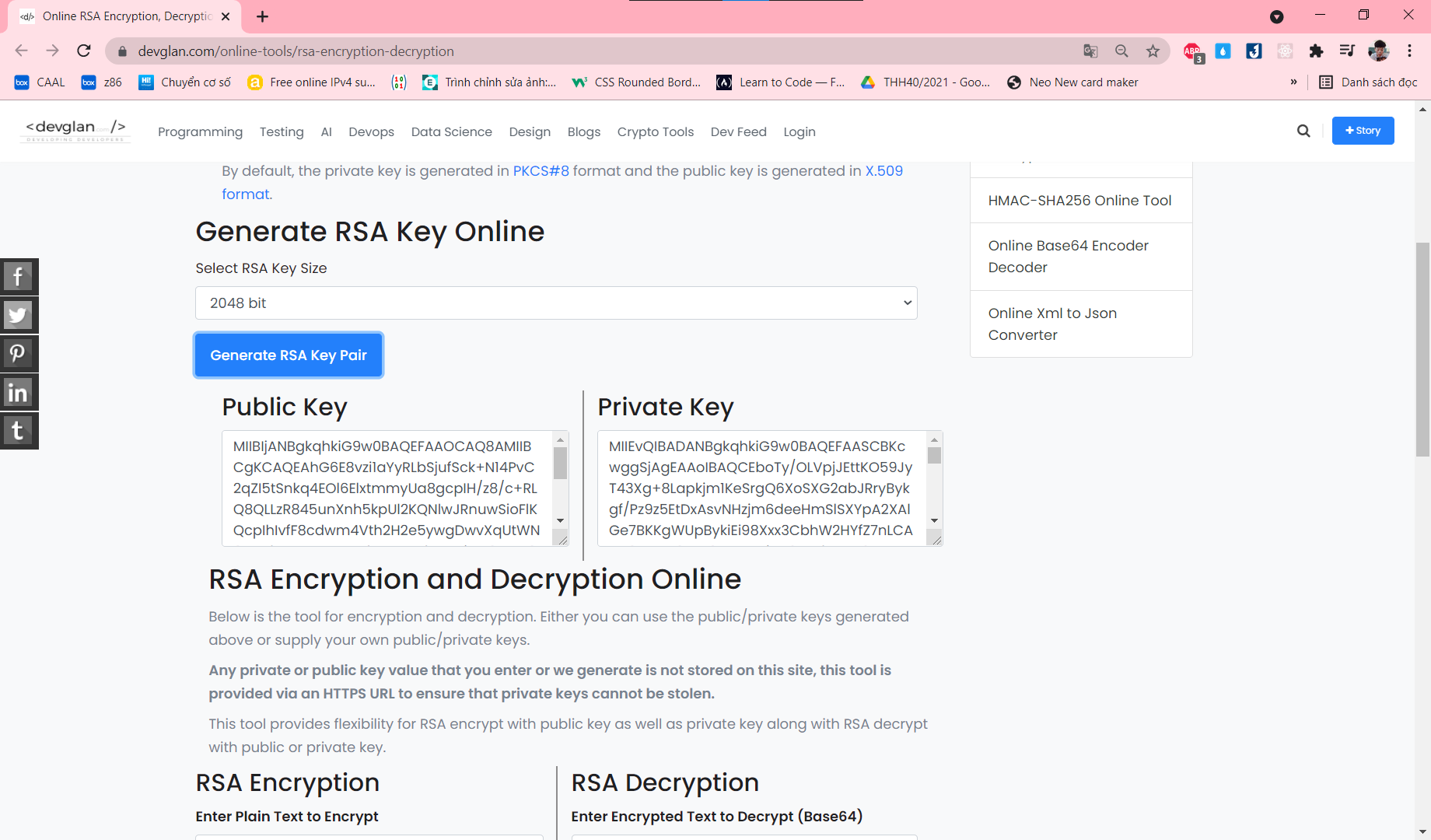
1. Tạo khóa public key và private key bằng thuật toán RSA công cụ ở web:

https://www.devglan.com/online-tools/rsa-encryption-decryption



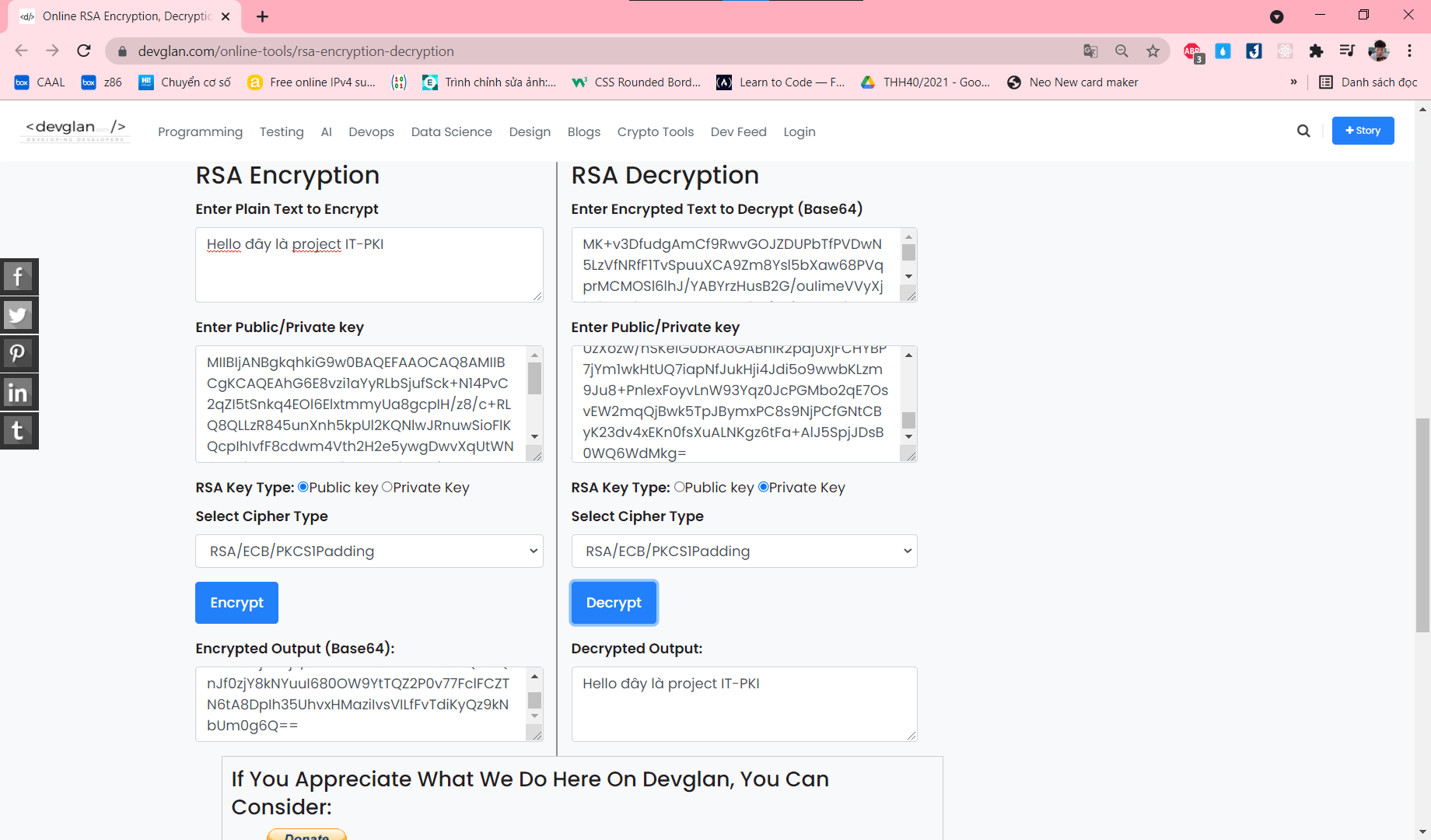
***Hình 2-12: Giao diện web công cụ tạo khóa dựa vào thuật toán RSA***

Ở trang web này khi chọn độ lớn của khóa ta ấn Generate RSA key pair để nhận 2 khóa.



***Hình 2-13: Ảnh khi tạo 2 khóa trên web công cụ tạo khóa dựa vào thuật toán RSA***

Sau khi có 2 khóa ta tiến hành mã hóa thử :



***Hình 2-14: Ảnh khi mã hóa dữ liệu dựa vào 2 khóa vừa tạo***