



BÁO CÁO ĐỒ ÁN CUỐI KỲ

HOMOMORPHIC ENCRYPTION – BASED ONLINE VOTING SYSTEM

NT219.L21.ATTN
GVHD: Nguyễn Ngọc Tự

Trần Đức Lương - 19521815
Vương Minh Tiến - 19522346



MỤC LỤC

MỤC LỤC.....	2
I. Context	3
II. Research Motivations.....	4
III. Proposed Scheme	5
1. <i>Homomorphic Encryption</i>	5
2. <i>Paillier Algorithm</i>	5
3. <i>Voting Procedure</i>	6
IV. Contributions	8
V. Comment.....	8
VI. Demo	9
A. <i>Main Demo</i>	9
B. <i>Timing Performance</i>	15
TÀI LIỆU THAM KHẢO	16

I. Context

Bầu cử là một quá trình tổng hợp phiếu bầu của người dân thông qua bỏ phiếu để chọn ra một cá nhân nắm giữ các chức vụ trong nền dân chủ. Tuy nhiên trong bỏ phiếu truyền thống, mỗi cử tri phải đi tới điểm bỏ phiếu và bỏ phiếu. Điều này gây khó khăn trong khâu tổ chức, cũng như khá tốn thời gian cho tất cả mọi người. Bên cạnh đó, những người ở vùng sâu vùng xa hoặc công tác xa nhà cũng sẽ gặp bất tiện lớn. Với sự phát triển gần đây của công nghệ, bỏ phiếu trực tuyến ra đời giúp khắc phục những hạn chế trên, những cử tri có thể bầu cho các ứng cử viên ở bất cứ đâu thông qua Internet. Việc bầu cử trở nên rất thuận lợi, chỉ thông qua một số thao tác đơn giản trên thiết bị di động hay máy tính mà không ảnh hưởng đến công việc, hay phải xếp hàng đợi như cách bầu cử truyền thống. Tuy nhiên, kết quả được gửi đi qua internet ở dạng thô (raw) rất có thể bị lộ ra ngoài và dễ dàng chỉnh sửa trong quá trình vận chuyển. Với việc sử dụng mã hóa dữ liệu trước khi gửi đi, vấn đề trên được đảm bảo nhưng sẽ làm phát sinh vấn đề khác là dữ liệu vẫn sẽ bị lộ khi hệ thống cần giải mã để thực hiện tính toán phiếu bầu. Trong đề tài này, nhóm em đề xuất giải pháp để bảo mật dữ liệu cử tri gửi đi bằng phương pháp mã hóa và đồng thời đảm bảo việc tính toán kết quả bầu cử chỉ dựa trên bản mã.



Hình 1: Bầu cử trực tuyến [1] [2]

II. Research Motivations

Như đã đề cập ở phần ngữ cảnh, việc tổ chức bầu cử trên mạng vẫn tồn tại mặt trái trong vấn đề bảo mật thông tin [1] [2]. Cụ thể, khi cử tri gửi dữ liệu đi qua Internet, họ không thể quản lý và đảm bảo được tính toàn vẹn và riêng tư của phiếu bầu. Server hoàn toàn kiểm soát được thông tin lá phiếu gốc của từng cử tri. Những kẻ tấn công hoặc tổ chức quản lý phiếu bầu không đáng tin cậy có thể sử dụng và sửa đổi, làm mất tính công bằng, minh bạch của cuộc bầu cử. Hơn nữa, mỗi cử tri có thể lợi dụng và thực hiện quyền bầu chọn nhiều lần, gây nên tính sai lệch trong cuộc bầu cử. Đã có một số giải pháp để giảm thiểu những vấn đề trên, ví dụ như mỗi cử tri đều mã hoá phiếu bầu rồi gửi lên server, họ sẽ cung cấp khóa dùng cho việc giải mã để tổng hợp phiếu bầu. Nhưng với số lượng lớn cử tri tham gia bầu cử, việc mã hoá và giải mã để tính toán thì tốn rất nhiều thời gian, tài nguyên và không thể đảm bảo việc che giấu thông tin lá phiếu (plaintext) khỏi server. Vì vậy, chúng ta cần tìm ra một cơ chế thực hiện để giải quyết vấn đề vừa có thể che giấu bản rõ với server mà vừa có thể tính toán để tổng hợp phiếu bầu cho cuộc bầu cử kín [1] [3]. Và Homomorphic Encryption là một phương thức mã hóa đáp ứng được điều đó.

Homomorphic Encryption



Hình 2: Homomorphic Encryption [1] [3]

III. Proposed Scheme

1. Homomorphic Encryption

Trong khi đó, Homomorphic Encryption (HE) [1] [3] đang được giới nghiên cứu quan tâm nhiều trong những năm trở lại đây. HE là mô hình mã hóa đồng cấu (nghĩa là, khi nhân hai bản mã bất kỳ thì khi giải mã kết quả sẽ nhận được bản rõ là tích của hai bản rõ tương ứng ban đầu) cho phép chúng ta làm việc, tính toán (phép cộng và phép nhân) dựa trên bản mã tương tự như làm việc trên bản rõ. Điều này giúp không chỉ tiết kiệm thời gian mà còn đảm bảo tính bảo mật cao cho dữ liệu của người dùng. Có 3 dạng Homomorphic Encryption chính được sử dụng hiện nay và chúng được phân biệt dựa trên đặc điểm và tần suất các phép toán có thể thực hiện được dựa trên ciphertext. Với Partially Homomorphic Encryption cho phép tính toán một phép tính (cộng, nhân) nhưng không giới hạn số lần thực hiện. Somewhat Homomorphic Encryption có thể thực hiện hai phép tính cùng nhau nhưng bị giới hạn số lần thực hiện dựa trên cài đặt. Cuối cùng là Fully Homomorphic Encryption hỗ trợ tính toán 2 phép tính trên số lần không giới hạn.

2. Paillier Algorithm

Dựa trên ý tưởng của RSA và El-Gamal, Pascal Paillier đã cho ra mắt Paillier Cryptosystem vào năm 1999 [4]. Đây là hệ thống dựa trên hệ thống mật mã bất đối xứng (Public-key Cryptosystem). Paillier Cryptosystem chính là Partially Homomorphic Encryption mã hóa đồng cấu trên phép cộng dựa trên dữ liệu đã được mã hoá [2] [5]. Paillier-cryptosystem được định nghĩa dựa trên 4 thành phần: tạo khoá (Key Generation), mã hoá (Encryption), giải mã (Decryption) và tính toán (Homomorphic Addition).

Key Generation:

- $p, q \in P$ with equal length
- $n = pq$
- $g = 1 + n$
- $\phi(n) = (p - 1) \cdot (q - 1)$
- $\mu = \phi(n)^{-1} \bmod n$ (μ is used as *private-key*)

Encryption:

- Plaintext $m < n$ ($m \in Z_n$)
- Choose $r < n$ dengan $\gcd(r, n) = 1$ randomly ($r \in Z_n^*$)
- Ciphertext $c = g^m \cdot r^n \bmod n^2$

Decryption:

- Ciphertext $c < n^2$
- Plaintext $m = L(c^\lambda \bmod n^2) \cdot \mu \bmod n$

Hình 3: Công thức của Paillier Cryptosystem [4]

Tương tự với các loại mã hóa bất đối xứng như RSA, hệ thống khóa của Paillier Cryptosystem [4] cũng bao gồm public key (n, g) để mã hóa và private key (λ, μ) để giải mã, trong đó modulus n được cấu thành bằng tích của hai số nguyên tố random đủ lớn p và q . Khi đó với mỗi giá trị r random trong khoảng $(0, n)$ sao cho $\gcd(r, n) = 1$, cùng với một bản rõ m sẽ cho ra các bản mã c khác nhau theo công thức $c = g^{m r^n} \bmod n^2$. Đó chính là vai trò của giá trị r , nhằm tăng tính bảo mật (security) và riêng tư (privacy) của dữ liệu. Quá trình giải mã cũng được thực hiện dễ dàng với công thức $m = L(c^\lambda \bmod n^2) \mu \bmod n$, trong đó hàm L được định nghĩa với $L(x) = \frac{x-1}{n}$.

Từ công thức mã hóa trên, chúng ta có thể thấy được thuật toán Paillier là một dạng Additive Partially Homomorphic Encryption, nghĩa là với public key (n, g) và hai bản mã của m_1 và m_2 ta có thể tính toán được bản mã của $m_1 + m_2$ bằng cách lấy $c_1 \times c_2$ mà không cần phải giải mã. Ta sẽ chứng minh tính chất trên như sau:

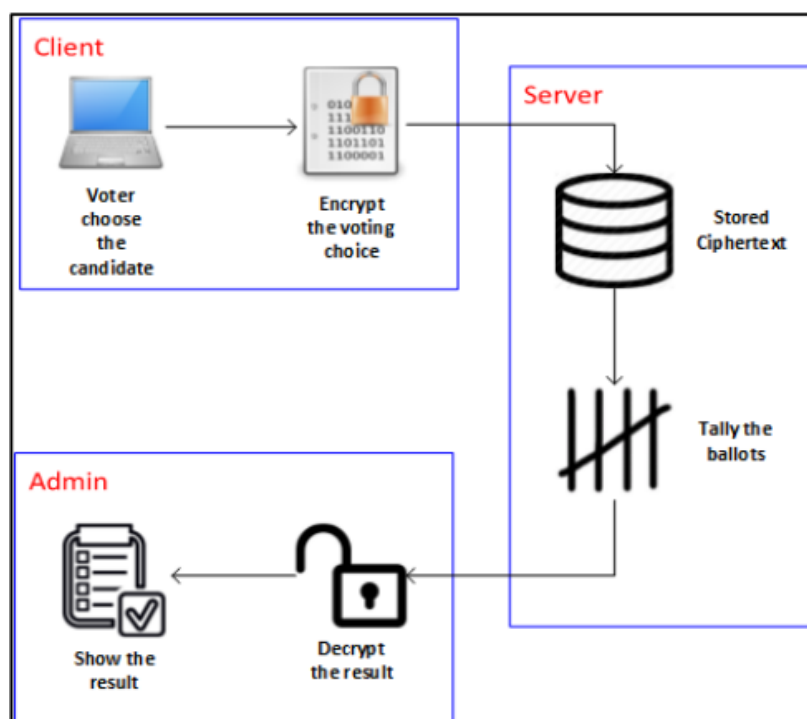
$$E(m_1) = c_1 = g^{m_1 r_1^n} \bmod n^2 \quad (1)$$

$$E(m_2) = c_2 = g^{m_2 r_2^n} \bmod n^2 \quad (2)$$

$$\xRightarrow{(1)(2)} c_1 \times c_2 = g^{m_1 + m_2} (r_1 r_2)^n \bmod n^2 = E(m_1 + m_2)$$

Tính chất trên rất phù hợp để áp dụng trong việc tính toán tổng số phiếu bầu trong một cuộc bầu cử kín mà không cần phải biết giá trị bản rõ của từng phiếu bầu. Đây là lý do nhóm em sử dụng thuật toán Paillier cho đề tài này.

3. Voting Procedure



Hình 4: Online Voting Procedure using HE [2]

Dựa trên những phân tích trên, trong đồ án lần này, nhóm em xin đề xuất mô hình hệ thống bỏ phiếu trực tuyến hoạt động sử dụng Paillier Cryptosystem để tính toán phiếu bầu [1] [2]. Mỗi cử tri sử dụng thông tin do cơ quan bầu cử (Election Official) xác thực để truy cập hệ thống, sau đó thực hiện chọn những ứng viên muốn bầu cử. Tiếp theo, phiếu bầu của cử tri được gửi đi ở dạng mã hoá dựa trên public key do cơ quan bầu cử đáng tin cậy cung cấp, với giả thuyết rằng việc sharing key là an toàn và bảo mật. Cụ thể trong phiếu bầu, những ứng viên được bầu sẽ được gán bằng giá trị 1 (dạng plaintext), những ứng viên không được bầu sẽ được gán giá trị 0 (dạng plaintext). Những giá trị (0,1) trong phiếu bầu sẽ được mã hóa bằng Paillier sao cho các giá trị ciphertext tương ứng không trùng nhau. Dữ liệu ciphertext trên được gửi lên server và lưu trữ trong database. Khi đó, server chỉ có thể tính toán trên bản mã dựa vào tính chất additive của Paillier Cryptosystem để tổng hợp số phiếu mà không phân biệt được các cử tri đã bầu cho ai. Cách tính toán của server sẽ được trình bày như sau:

Đối với từng ứng cử viên, server sẽ lưu bản mã T là tổng số phiếu bầu dành cho ứng viên đó. Tại thời điểm cuộc bầu cử chưa bắt đầu, giá trị T được khởi tạo bằng bản mã Paillier của giá trị 0, gọi là c_0 . Khi cuộc bầu cử bắt đầu, cứ sau mỗi lượt vote, server sẽ áp dụng tính chất additive của thuật toán Paillier để tính toán phiếu bầu tại thời điểm hiện tại của ứng viên đó. Cụ thể, quá trình tính toán được



chuyển thành công thức của dãy số, với T_i là số phiếu bầu dạng mã của ứng viên sau lượt vote thứ i :

$$\begin{cases} T_0 = c_0 \\ T_i = T_{i-1} \times c_i \pmod{n^2} \end{cases}$$

Cuối cùng sau khi quá trình bầu cử kết thúc, kết quả bầu cử T của từng ứng cử viên sẽ được gửi về cơ quan bầu cử để được giải mã dựa trên private key nhằm gửi kết quả về cho các cử tri. Quy trình bầu cử và xử lý kết quả đều được thực hiện trên bản mã thay vì bản rõ như cách bầu cử truyền thống.

IV. Contributions

Đồ án này nghiên cứu về mô hình hệ thống bỏ phiếu trực tuyến sử dụng thuật toán Paillier với thuộc tính additive HE [5] của nó. Mô hình đề xuất mang lại một giải pháp cực kỳ hiệu quả trong quá trình tổ chức các cuộc bầu cử so với cách tổ chức bầu cử truyền thống. Bằng việc mã hóa phiếu bầu gửi đi, dữ liệu của cử tri hoàn toàn được bảo mật, riêng tư thay vì bị toàn quyền kiểm soát bởi cơ quan bầu cử ở phương pháp truyền thống. Bên cạnh đó, khả năng tính toán kết quả phiếu bầu trên bản mã của thuật toán Paillier HE mang lại tính công bằng trong cuộc bầu cử kín, tránh những trường hợp cố tình sửa đổi kết quả ở bản rõ như trong cách tổ chức truyền thống. Đồng thời phương pháp được đề cập trong mô hình này mang lại hiệu suất cao bởi không gian key lớn (2048, 3072 bits), gấp rất nhiều lần so với số lượng cử tri bỏ phiếu trong một cuộc bầu cử.

V. Comment

Ứng dụng Homomorphic Encryption (HE) trong bầu cử trực tuyến có thể giúp bảo mật phiếu bầu trước khi gửi đi, đồng thời mang lại khả năng tổng hợp kết quả dựa trên bản mã; điều đó giúp tính trung thực, công bằng của cuộc bầu cử được đảm bảo khi trải qua các khâu trung gian. Tuy nhiên bầu cử trực tuyến cũng gặp phải một số khuyết điểm mà HE không thể xử lý được, chẳng hạn như việc xác thực một người dân đã từng tham gia bỏ phiếu hay chưa. Đây là vấn đề cần giải quyết vì nhiều cử tri có thể lợi dụng lỗ hổng này để thực hiện bỏ phiếu nhiều lần, gây nên tính sai lệch trong cuộc bầu cử.

Vậy nên với đề tài này, nhóm em xin đưa ra một hướng khắc phục sử dụng công nghệ Blockchain [6] để xác minh danh tính và quyền bầu cử của từng cử tri.



Trước tiên, thông tin cá nhân bao gồm cả chứng minh nhân dân (ID) của người dân sẽ được các tổ chức chính phủ cung cấp cho ủy ban bầu cử, cụ thể là danh sách những người dân đủ tuổi và đủ quyền công dân tham gia bầu cử. Ủy ban bầu cử sử dụng lượng dữ liệu này để cung cấp cho mỗi cử tri một ID token [6] dựa vào ID của người đó. Những ID token này là bằng chứng bầu cử của các cử tri, nó sẽ tồn tại nếu cử tri đó chưa tham gia bầu cử và sẽ bị xóa bỏ nếu người đó đã tham gia bỏ phiếu; Chúng được kí và xác thực tại ủy ban bầu cử. Sau đó, thông tin cá nhân kết hợp với các ID token sẽ được lưu vào một permissioned Blockchain của hệ thống bầu cử trực tuyến, có nghĩa là cử tri phải cần được phê duyệt để sử dụng Blockchain này.

Tiếp theo, quá trình bỏ phiếu được bắt đầu bằng việc đăng nhập xác minh của các cử tri. Để truy cập vào hệ thống, cử tri cần phải sử dụng họ tên và chứng minh nhân dân (ID) của mình. Sau khi hệ thống kiểm tra thông tin hợp lệ và có trong Blockchain, nếu như ID token còn tồn tại thì cử tri đó được thực hiện quyền bầu cử của mình một lần duy nhất. ID token của cử tri đó sẽ lập tức bị xóa bỏ khỏi hệ thống ngay sau khi quá trình gửi phiếu bầu hoàn tất. Khi đó, ở những lần đăng nhập tiếp theo, cử tri này chỉ có quyền xem những thông tin bên lề và kết quả của cuộc bầu cử mà không thể thực hiện quyền bỏ phiếu thêm một lần nữa.

Trên đây là đề xuất cơ bản của nhóm em trong việc hạn chế vấn đề bỏ phiếu nhiều lần (double-voting) nhằm tránh gian lận và mang lại tính công bằng cho cuộc bầu cử trực tuyến.

VI. Demo

Tại tài nguyên có hạn nên chương trình demo này không thực hiện trên mô hình client – server mà nhóm em chỉ mô phỏng lại trên máy tính cá nhân nhằm kiểm tra tính khả dụng của Homomorphic Encryption. Dưới đây là link github (bao gồm source code và cách biên dịch chương trình) của demo:

<https://github.com/mt2651/Homomorphic-Voting.git>

A. Main Demo

Chương trình có 2 hàm chính là **Vote** và **View Result** để thực hiện mã hóa, tổng hợp và giải mã, bên cạnh đó có hàm **Information Key** để mọi người có thể thấy rõ hơn được thông số của các key.



```
(kali@kali)-[~/Homomorphic-Voting]
$ ./voting
1: Information key
2: Vote
3: View Result
0: Exit
Enter mode: 1
```

Hình 5: Giao diện chính

```
Enter mode: 1
p = 425291489150439070060136370134980800699633090325748327024747738688052783528143054334075978686611466146667610434574285068192431490727763438213326814166166
47498913115032101548223886077937426893972667156197403952844953965274902015257987890884406404962618794737056038636668856700633737619851764367103785353504000
543394920504676855945646152367262651510584715371020038716781668423531096520367064800363390839703423600105402418528852599636940041197331897713965720441403
q = 390441262856364107930276575833004022556802142102115039321187918041678593810341431156651703004326057013588532542180327630586090108923978841864843947172788
0320731038876747386355268611650083904201448927716712358857272579987776358966158045272458961729093932088724496257766351640615699877782526715855265085301870
547763393274454200476471972841071434176898196274569521223951616665523815195782921020771037072049102008433038323395799242212701379861882906034200618647917
Public key (n, g):
g = 166051346105961105085040713395349772143016556621929931266458632575430545073255091646904212137836728124947425750052066391381550849972066361010406722161414
1589732739104511095336817329937398360037648657047487624355903026151023044490410789745562836287725975544052452909240499193665797418162678979649991003807919585
7188970601065728812830502693793507987576088818494799973754856214180903911153585453917523117561503744252128897307466298562155228312227825973374715639954057009
8266966723879432392010428797123385848417806196999399368344195491580694002316199233422483958159628632784400262314898784147484548567489879583117895783016259306
478185677324705537619072958270565099040651502637377567635640100547949748608428891529182127414645607107820903661925874447297328300609747395935831299110796
162247680549263295819545129156487798170542902745695379743327184312149363510306667599523877749042545357426788209930238539126398168743078286507552
n = 166051346105961105085040713395349772143016556621929931266458632575430545073255091646904212137836728124947425750052066391381550849972066361010406722161414
1589732739104511095336817329937398360037648657047487624355903026151023044490410789745562836287725975544052452909240499193665797418162678979649991003807919585
7188970601065728812830502693793507987576088818494799973754856214180903911153585453917523117561503744252128897307466298562155228312227825973374715639954057009
8266966723879432392010428797123385848417806196999399368344195491580694002316199233422483958159628632784400262314898784147484548567489879583117895783016259306
478185677324705537619072958270565099040651502637377567635640100547949748608428891529182127414645607107820903661925874447297328300609747395935831299110796
162247680549263295819545129156487798170542902745695379743327184312149363510306667599523877749042545357426788209930238539126398168743078286507551
Private key (gMu, lambda):
gMu = 139927797113337532828306792392273594053634268888309841109780357666439199857488731339968323503230520301555135296565613185527718577884591015777892415160
39472976473816871964409784225630151928992236387792522336882075272993681026250080017454119186237973945413397498298367432724347466192643527004178700177391308
68323213803648605523064912324385749323807758547706720485187387819108159445408573625502890031144681262088307173926593141876394096333011792888004735030709083
28655326452084263359746385768164539582661893025541210059876273620421974259800174382274335139892005231323272530798635311532894550923147393652388570636573377716
6690294345782527460663788067574793421559405740938646305453305771959854306210745484568238736294480011652758402008016536875847005547693803369071113023216041
04899558208524294316775018489097762808533609372305633082909173026943327030702112353759487265455954327724756673216212440082080208369871582305335203
gLambda = 8302567305298055254252035669767488607150827831096496563229316287715275366275458234521060689183640624737128750260331956907754249860331805052033610
807079486636955225554766840866496869918001882432852374381217795151307551152245205394872781418143862987772026226454620249596832898709081339489824995019039
59792859448530853284406415251346896753992788044409247399986877428107090415055576792726958761558780751872126084486637331492810776141561139129866873578199770
2442624958030213808166753579150993205301024472817053313823873802612592214800769338250803271534322831563035805129342681445314766229160663893583020880943827
7654151213841419506319146481456583848339713036434875479799439356437110868628014623430127753464287343909913286462927578925669003616048904489087844617376220949
97482392167284535757416493558602938697530529919733758110256172108465702671824639330865263077198697777414243712331840408298559033591682497455973709116
```

Hình 6: Thông số Key sử dụng

Đây là bộ key 3072 bits được tạo và lưu trong 2 file ‘privateKey.key’ và ‘publicKey.key’.

Tiếp theo là hàm **Vote** với 2 quá trình là **Encryption** và **Addition 2 ciphertext** với số ứng cử viên candidate là 5, trong đó ciphertext thứ nhất là kết quả bầu cử dạng mã tính đến thời điểm bỏ phiếu được lưu ở file ‘resultElection.txt’ và ciphertext thứ hai là phiếu bầu dạng mã của cử tri. Nếu file ‘resultElection.txt’ không tồn tại, nghĩa là chưa có kết quả bỏ phiếu trước đó, ciphertext thứ nhất là các bản mã của giá trị 0 tương ứng cho tất cả các ứng viên.



```
(kali㉿kali)-[~/Homomorphic-Voting]
$ ./voting
1: Information key
2: Vote
3: View Result
0: Exit
Enter mode: 2

Not exist file 'resultElection'!
Start new voting
Number of Voters: 1
```

Hình 7: Bắt đầu bầu cử

Đầu tiên, để có thể demo với nhiều cử tri, chương trình bắt nhập vào số cử tri tham gia bỏ phiếu, sau đó là kết quả bầu cử tương ứng với '1' là bầu, '0' là không bầu. Ở lần vote dưới, nhóm em điền số cử tri là 3.

```
Enter mode: 2

Not exist file 'resultElection'!
Start new voting
Number of Voters: 3
Rule: A voter is just allowed to vote once!!
Election is ready!

Voter 1: 1 0 0 1 0
```

Hình 8: Cử tri thực hiện bầu cử

Sau khi nhập vào kết quả, chương trình sẽ in ra phiếu bầu của cử tri sau khi được mã hóa.

```
Voters 1: 1 0 0 1 0
Encrypted Vote 1:

Candidate 1: 14353042262628199231092453537140466285758865314152010094359650914710773306279987453415237373605076068640494211169665756332395818938554233236069
9459895639230016209990355191036408005763200007517788243823576085190102140140775720363152392933554913680467249530700849157196153548604664297569074096457359527
093077914799486750796835139094747303557402941475676773329677559002890969844189611190977370485948298495905050046112945212645807930370929709362486033536956417
638252486000233003835000498595618188496997224180624992087532462305227281860141977661067824098934914609593752927311498525453647339111020633053391396486280453
4672632860412335821703577519116569554059799231776398756129072826208300480684873891255872710474136619464280416507408014972614304567263894059534906178832257849
0002873647776804662617651341611356075902244595858785583453878721021062032526214874525846016493583122723698412246646714707920069588280650581937612630573647
94281617598518324106230484427799135098804997944741409128052118001174374635104425944123797835868816421044832489430093106543408536971948239072074257912903005
690017233985068693827689775183784655710747973361930006088714879922064720788818717667875177927944554336180720752628865183232764802974912917255299095013991345
978224817321283279144388887944649465111429779219295349745981734171765980934663484829375456798973066058578290108287676175959736204835274803518235919537895147
3314296155020960951593314134199818073985619844804969340268733847156105461562523936462749478822437322322741389001385319066014596803427867051694951127980641076
00981815920622352794942309373036556272583900231658204221666847340723047416286986422525301037338351241514434903599276562299977805709447577900338389656400211
96651295935439815229716864432189475000066639462073100009447992551756127384045135736161480906458072951161287913889642750901894628584253820

Candidate 2: 1806835234569368863598235285347015829127345012302409933915715405703504528014767560896395224685689925393984871439021884753897463690194141207309
67614259780354731049292239673292099512509663789240169776070167335259580645608172390511694861490489340963114321366934713933968208764737837618195874816228369
923100949919888347368506711306752776268450611149298491317070793747819436391150422819134311296756221101176110372416828076988088343428592644943979626899307422
41706055240155881687471525813835533847958175919057480236198116140114833104732874028604852770893489837365044167138266449946942341834223303540763892747667419
4119482060693119577019103492347597076348754512452817954518524600628793349046600259385605607545661223611096965162594782187133369343374963391272069905951600646
798932129172336545627372304769299572596712069859426143885495977889048195055765225135252404677554076863004046147792338196617127275908478782151258889632527905
0529717581695213472435631737906901923573239466905325492668658833698063670848602014429676086472129709134501857041354733596561998879298329740476153019648727268
6190105442332803027566407781099044593236403658939144620157272341798496024210754139202540178181081028017800211263334821878313028423944632393273073196026242559
066160047897106754068803520812757093705883118080893071927135678530818008062834622415231846330897703941954615420026608808855601504747939963447838718348902201
7290135188376605027786254371667483077182939085826268515326599115359143670435178606553476394957602824160734300762803122622409948092113443687046427206187453910
00904786483024396807729010637860434885512619731593373096362381458092818054731705849928589274151230175403796812392359019358112081428650912967427947135666261
70988368687165471629184659189679571153985036441523702767056746909671439285088530909533171656457540067621706246248923240048014573629812266
```

Hình 9: Phiếu bầu được mã hóa cho từng ứng cử viên(1)



```

Candidate 3: 1255832619305296238861500382636985000780166658861179361862953840514110751405208832246713819260884818525905709099019648645076590506360892650403
035727587113230610963134266427891177739499811703527132233273946389337957599931705690803934027236884300083856297992287154852355882781005459488378805789690
0643978644113842642381843223837600949977372798070004749107757504033047140125711269243553383045376048500326516471853866963504256369958180186162924386748134
910019233992384769006094475099974744297658364519010246216268241586174950689629738243138881835189683255588177065532353487531000568032588132834519520978754
450843609723236737415920533429351814552663448848677259111521680916405043450103706371605522775143276580080520601440461962199764279335849842779067210929734
822724469943562161942965256425465672894709813179540183918627510483408004758894075522745685810981588435851976258232434337896177148105533770120625084665731654
783693710348200806148920841539099536432476843473395323610935306330294399626380889809659855176921436271319896086121208142953833777864534265410604631367
1639863612328331029801670904632619191955964889311471477689703856970053183381035282191198975133247485873830023985501326364814468086969789728179878939140
0281630441271015862850909438394015149476958568343220438229427981518510648771827593871124254796716495822005414618802089041816994135798478186871821818655468
27627192494210016990255269932720562467102745343826977244112807241324826880264415290336021854235123241089171464489362474196556317013667948550653496325072720
31174171033998054974390178458232994910887586585244749624234829414376889192659241964630222854296869247866955926792129880084141034136523432738882196327620065
8776418921388049196256378776275340812444988665678513068062520463284798807083612019497229426600002709370270945776452223968024646591410284

Candidate 4: 18251586430991180780059017015802211403378323199660255213070950501521194466965009823051385317623600941009637039354968923236418042543952473947728
511034188872996941324024278222536505103193850719820704525461074717243879158697062880427978726147715388093144161807499002690235709377265443814127497299845561
1552549420957785229012405673405644845663160506671692402240913521311124286019370951832549055633970040814725428445917276766606302499959625185520678872686520
20562814187945170234243553937082893392491499964006744130894002430221773396001670955212323881424995513407150461525184880874010416057387381996909821255621
781395937029382296251932452918847638480300252654118690645810434740977348291066704169802053941590741698902890532087509678073402563633850717625753975280
123532328181495438906464743609957612700528354302505491382253788799578687170511944225694458397451548392502079033738850264588821994151482862806724876240225
469759140645462968468805760913772930705505440023764385677780561193669350938335135166896681804206840935164738338973438839246282579015332910468277120810172526
04580369290108131910367840769683838355783277622434463531574899822039427562816638066160461114682421495551100520532480999381645064190207740645110594258586052
9803440498983071100701917370306045873158887145830609291415656693277088997522408116139768706702018067814765383715831887259266123258944063070503999846103146
80364769640200616769117634260212953697690427161521657320773651679903628715423550459245711897793968468001830103306152822611956077413024933812204793231581
2293551329992742960896398269838541109769369438504005203948825836822662713410943891037431965756060219185083390408173650686029181531852431562063584299274090
80771731594051048668189627882959752166647694442866745466382864551866724459380483528279715818137013276793663648179294456072721208134507

Candidate 5: 1345339398698403047073775209982270036649887911343466343320638722962488607387462995653664701883123068115617515682506383585071963346845119719851
52224979616678398994075421881433828568448647953452413245641354585688634212175212680243765973646714985098368025544382525607641869670612056848597081923695
2171654943049418655371291374040402564823806964366461677291996099880685748642066926385377043653055627322196174022493103047355761508091367751453430066490186563
47740047837805474911353868205093366278512949997286280998797323853485371266652162936356916028649151325884934828784772285627362748521275747948780433486874457
8341139296232708525407648247694024196294966256421456198113633129080608360455782015982223350307175280483942331675871468693719638713732588008783113076301784
1334551685164930375359944894542800366722841952182295236124717216088683811554522923774774049023894678193009598384372780912168328463355956438061867799296409
763053168280736800746130090352470779333044181933636776715416092851927118786305042236806780055231545827805797812877299638651135793408438226490457052156145
828789226014003519041051127584201874580606099483947688231801845801572967163020160962510621129270947853861980939655734785657488765860283292177453403236
42765384896063248571278215913169442649606176155167970151576805837605112677853165765252389198574933862369140815885039785986810792468938876260526586636963715
50076268165803522604547404002169209466457680801457617619295757947298765391486672341909630126004472834699483391850742410072796210692326210972422000
3516571109391211880516564698405219131052572990580438468756195666802817848091718300070043772623784823536991873468803596602924521390537195938592218354556
68645466323088343736185184645541086291416535949078857291078860357781825847722865403968340515085324827943828411858530278264409764904563319

```

Hình 10: Phiếu bầu được mã hóa cho từng ứng cử viên(2)

Chúng ta có thể thấy cùng một plaintext là ‘0’ hoặc ‘1’ nhưng đều cho ra những ciphertext hoàn toàn khác nhau và không thể đoán được quy luật.

Tương tự cho 2 ứng cử viên còn lại, với mỗi lượt vote, chương trình sẽ tiếp tục **Encryption** và **Addition 2 ciphertext** đối với mỗi phiếu bầu của các ứng cử viên. Sau khi thực hiện các bước trên xong, kết quả cập nhật trở lại vào file ‘resultElection.txt’.

```

Candidate 5: 255826184449932428615251676813305603125731476001642219918487873184721458093244960769936549532451991163028578610838777634156388218833105202274688
842300477918428049493033692860898012039769768648047662626471565028118141640058869929335287034140198616863618313697817932434989459488497105347817159741009
621965123339362341628833964212593458723127126468769119026106354677920628394822502548435496988216614063831245347024559661294565878812042229269286086654803
05021339719411784361854396417132739689071665991581799140762524387053675869908292025535808778895208836554251741697124005743729043843364809268893137523366044
750032777929912916349664905617720237558068057808464578351653243212698523818425815632018293277185104627119003108130866564561533596489552818882904100123308299
873185379881369785898640447031597565089396693645671472565537203873840587229855343237518742546854380618793260200530832647811948413471210911090097437565998833
0236084270307402049024015102539324549060381560847480070644254333014409855116421713768374538304261266021657723220298870224988253655679974290077056141263633
70976559358943890083487863589497344232864473719017711438426800573294219046501269492834418286309468273814045038261383478358166461361793147272557516141669300
0473872963735287252609760103676242070338680915830976788039093550032908086396480349269732934616885463574034829848208264914579721779151121657043263894
7576439765601499256720810439644502199794254013601039275274508178864382407096563289772629714752874149511007458005570618641286589895389648091463
20581063987234254394590853591760046147081413005814614620659895393782958307465136708827968524543720243126363548751013145355357649415611970618538895529674994
79504566180142447510208977600686286957224623340778510663540690459918394290679753198988666297152948370218884719476749987430400722936639356

The election is successful, the results are saved in file 'resultElection.txt'

kali@kali:~/Homomorphic-Voting$

```

Hình 11: Kết thúc lượt vote

```

resultElection.txt M X
resultElection.txt
1 2168305299129520728693302792286219754772793249749987069883702934509294401736636374406903054039958925239863870048480945822747346441
2 260264851167745289208395354087343406955504472151048169601639038980695786944682274356254541223401965475452819034227455321833378964
3 144437853314273934350861247673911637244621878772845447594507762668760209251849829829450205544022442506632059710418170337450089464
4 13446449546890206896315727852328105249692599207574686450210576539167417815001193353562653574578759582014368623433109921919531511
5 23287045144856150567655961440340084286294286229849136000330825678882406433220520631307320994449709897215802134073940766995391917
6

```

Hình 12: Kết quả mã hóa được lưu tại file ‘resultElection.txt’

Mỗi dòng tương ứng với kết quả của từng ứng cử viên sau khi tổng hợp. Vì kích thước key là 3072 bits, nên ciphertext sau khi mã hóa tương đối lớn.

Tiếp theo là tới hàm **View Result**, đây chính là quá trình **Decryption**. Chương trình sẽ lấy kết quả được lưu từ file **'resultElection.txt'** và đem đi giải mã, sau đó in ra màn hình cả ciphertext và plaintext.

[illegible]

Hình 13: Xem kết quả bầu cử tại thời điểm hiện tại

Có thể thấy, sau khi trải qua lượt vote gồm 3 cử tri trên, kết quả bầu cử đã có là ‘2 - 2 - 2 - 3 - 1’, tiếp tục thực hiện một lượt vote với 1 cử tri để có thể xác nhận tính chính xác của chương trình.



```
(kali㉿kali)-[~/Homomorphic-Voting]
$ ./voting
1: Information key
2: Vote
3: View Result
0: Exit
Enter mode: 2

Number of Voters: 1
Rule: A voter is just allowed to vote once for only one Candidate!!
Election is ready!

Voters 1: 1 0 0 0 1
```

Hình 14: Bắt đầu lượt vote mới

Sau khi thực hiện mã hóa và tổng hợp, kết quả được lưu tại file ‘resultElection.txt’ đã được cập nhật mới.

```
resultElection.txt M X
resultElection.txt
1 2521075176068037474009643753993018593657346500274093585611042089550619925955915631736422388751308495909468294145872047850599811381
2 352691232090430385015288414088476629612758824651889921193799201560342033104343563142834337653057503435621568866560618096012137782
3 110529961971374064651868941482851205734412252105719221814808170739740475332919835372032627501303622985162588156902253014377471271
4 227376630466263682965221719423581033806950425491914324618476323911976878597428494288847249615027893087368749794192526737713419133
5 657301891673744070514277453757979823423393546856850476984840162451286632638030503820458000791874582349948933109780324104919445169
6
```

Hình 15: Kết quả được cập nhật vào file ‘resultElection.txt’

Và thực hiện giải mã chúng ta nhận được một kết quả chính xác.

$$‘2 - 2 - 2 - 3 - 1’ + ‘1 - 0 - 0 - 0 - 1’ = ‘3 - 2 - 2 - 3 - 2’$$

```
Candidate 4: 227376630466263682965221719423581033806950425491914324618476323911976878597428494288847249615027893087368749794192526737713419133656364289658512
7299003896587243929188147720168840118585556732375903323138546675181821501360235296037627533239609408844056977425603544888743039747950965116888178726275416
1713650664922075870871017989012960943849583447663821052360424485836970170189524215905071372735546207444439666613970468057701617706529272085652427405477011229
366908489904070068168015117518991193303360052521784249958978795410449268501013054719300790714856113470537802599247386399806975116052535554604292884218355047
9618920108721452528359843358117135528582172951495952590599124594041789746710410890413864852410262022642371115575994437700095731713572211494864345891462488346
3255615908702642891891981173676689573454865962702243433488517251270621417085953266697728004578127906793006227693485291091383004976784150186739169364594804
40809532467811017327118583227959189511659102942119100204424042629644130424593063305305345251645437875832591605504587664619131720972623367806046263776668850094
5886771541981541082554860477257112692738165106715902930140880883891762821407319008442485616153718901714575696885182453286439514722609310071786428229295583
0612146309768897221039721659919929326512215741393011490157785422021534368168224119300642421105186317831212266975392669395022243055050344355123935571329827573
1754149851212498010746270017347916999377474521003750002637351517826919483977161478011331296692062644450435619336493122231344762004656143851680
1623154178304322916532812975814080995934259652929762362032117834036563048109140309058835392801508389740949984211006357324519282331408827849137349529315936601
89409325191102457208081576004922739645568300805375638121203560460184021119729042140575700624692672531852421713618289832036964243127881408

Candidate 5: 657301891673744070514277453757979823423393546856850476984840162451286632638030503820458000791874582349948933109780324104919445169636643353098529
89043532706594849963730073601532076132472319785723132693669080407608212325116037647888194962528830878880240731952102029750173363657699291853108993903991726
757204448060808724920271130714861368516111151298782337180810956152148738906537331855624846741743637239560625518592154102493569558122686869041279107678881351
01935344202371345352832134411980126291557708408766855085401458901393737456302375275818039601045668693337666091112956763741220602264011963775026055499962658
205116582507708532707939149737555135172843328606618945631877486842573408409825009261220069192363242595685672996980433280783350901720466827980697683156366672
8807473144928272048295019643759226928824100936664912417549322184716401703255373193092785436313670172539882867005129899101400746053204830730545156722182970459
9250833482444757296396689962542745116900125766638424002124004816812069487203496879873329359838684242399843467601783897005297903387477967303574486542957181487
545478761147254392122920671022157956952741098147209865530942159567321687925500816609013740704092096974609482922088185194188159813045553861599469202704882817
43264800686838679731654649622978527701686461632511242814771004479618274573173595112349239836867855044889409143174558875182267429106395481871344212085009531
942552198863661808887878363243476604091379062950838095661504328630512369880591598804789379232563710518621895996798702692837870489776474928137474037585152393
051265222813144667255795574402565538055858118743568443026080291690807840246435169970164540782796724159727347903130520152981536102360159743601960176384470
191352302895818425966817377630790836747021906307989661256463935760601506151921856894676326061227171418548908247473562000940529425696400

Final result: 3 2 2 3 2
```

Hình 16: Xem kết quả bầu cử tại thời điểm hiện tại

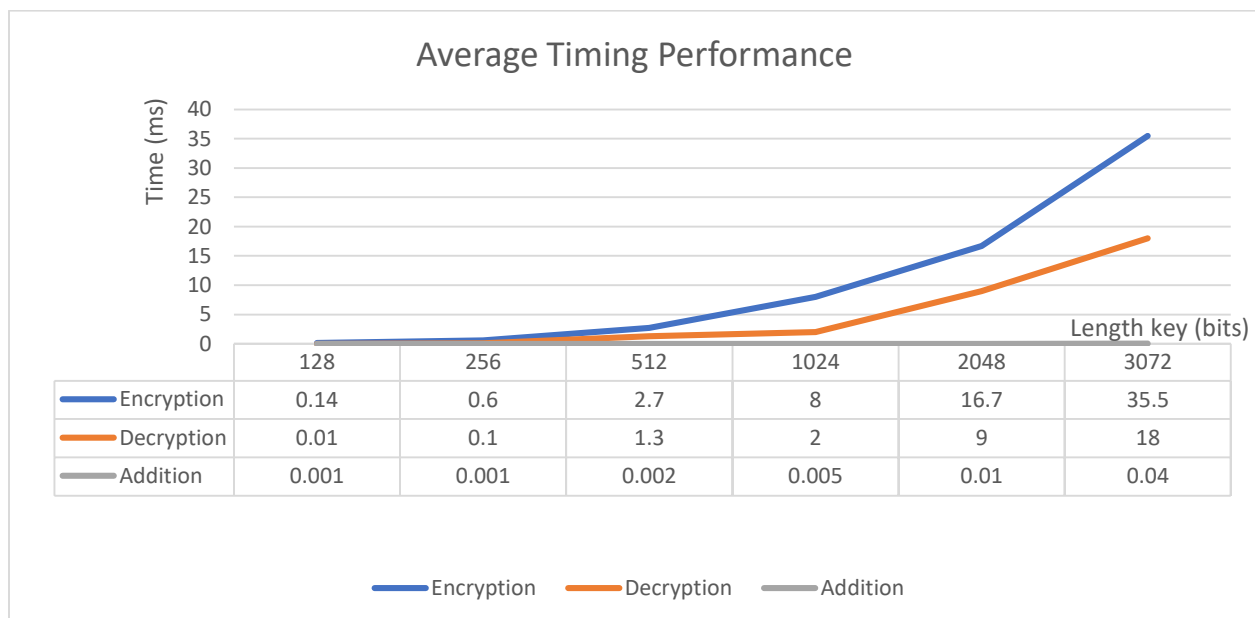
B. Timing Performance

Bên cạnh việc tính toán, nhóm em có thực hiện một chương trình giúp hỗ trợ việc tính thời gian của thuật toán và chia làm 3 phần: encryption, decryption, addition.

CPU:	Intel Core i5 Coffee Lake, 8265U, 1.60 GHz
RAM:	8 GB, DDR4 (On board +1 khe), 2400 MHz
Ổ cứng:	SSD 512 GB NVMe PCIe, Hỗ trợ khe cắm HDD SATA
Màn hình:	15.6", Full HD (1920 x 1080)
Card màn hình:	Card rời, NVIDIA GeForce MX250 2 GB

Hình 17: Cấu hình máy

Với cấu hình máy như trên, nhóm em đã thực hiện đo thời gian trên bộ key 128 – 256 – 512 – 1024 – 2048 – 3072 bits với 10000 vòng tương ứng với mỗi phần. Dưới đây là biểu đồ thời gian trung bình của từng phần (1 vòng):





TÀI LIỆU THAM KHẢO

- [1] S. Saproo, V. Warke, S. Pote and R. Dhumal, "Online Voting System using Homomorphic Encryption," in *International Conference on Automation, Computing and Communication 2020 (ICACC-2020)*, 2020.
- [2] S. M. Anggriane, S. M. Nasution and F. Azmi, "Advanced e-voting system using Paillier homomorphic encryption algorithm," *International Conference on Informatics and Computing (ICIC)*, pp. 338-342, 2016.
- [3] B. Raut, M. Jagtap, S. Ghule, K. Jadhav and P. S. P. Aundhakar, "Homomorphic Encryption Based Online Voting System," *International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET)*, vol. 6, no. 6, pp. 159-163, 2019.
- [4] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *IN ADVANCES IN CRYPTOLOGY — EUROCRYPT 1999*, Springer-Verlag, 1999, pp. 223-228.
- [5] T. Sridokmai and S. Prakanchaoen, "The homomorphic other property of Paillier cryptosystem," in *2015 International Conference on Science and Technology (TICST)*, 2015.
- [6] M. H. A. Z. A. & I. Z. Murtaza, "Blockchain Based Anonymous Voting System Using zkSNARKs," in *International Conference on Applied and Engineering Mathematics (ICAEM)*, 2019.