Kỷ yếu Hội nghị KHCN Quốc gia lần thứ XII về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR); Huế, ngày 07-08/6/2019
DOI: 10.15625/vap.2019.00066

ỨNG DỤNG BLOCKCHAIN ĐỂ TĂNG CƯỜNG TÍNH TOÀN VẸN VÀ BẢO MẬT TRONG QUẢN LÝ LƯU TRỮ VÀ CHIA SẺ DỮ LIỆU IOT

Lê Trung Kiên, Phạm Thị Ngọc Mỹ, Nguyễn Hoài Quốc Trung, Phạm Hoàng Anh*

Khoa Khoa học & Kỹ thuật Máy tính, Trường Đại học Bách khoa, ĐHQG-HCM

{1511640, 1512049, 1414294, anhpham}@hcmut.edu.vn

TÓM TẮT: *Nhu cầu chia sẻ dữ liệu có sẵn để rút ngắn thời gian và tiết kiệm chi phí trong quá trình phát triển và triển khai các hệ thống thông minh dựa trên nền tảng IoT ngày càng nhiều trong thực tế. Tuy nhiên, việc chia sẻ dữ liệu trên gặp nhiều thách thức về việc đảm bảo tính toàn vẹn, bảo mật và công bằng trong quá trình chia sẻ dữ liệu. Bên cạnh đó, các thiết bị thu thập dữ liệu IoT thường bị giới hạn về khả năng xử lý và dễ bị tấn công trong mô hình tập trung như hầu hết các hệ thống IoT. Trong khi đó, công nghệ chuỗi khối (Blockchain) nổi lên như một giải pháp tiềm năng hỗ trợ giải quyết các thách thức trong quá trình chia sẻ dữ liệu IoT như đề cập ở trên. Trong bài báo này, nhóm tác giả sẽ trình bày một giải pháp ứng dụng Blockchain để tăng cường tính toàn vẹn và bảo mật trong quá trình lưu trữ và chia sẻ dữ liệu IoT. Kết quả hiện thực thử nghiệm của nhóm đã chứng minh tính khả thi của giải pháp đề xuất.*

Từ khóa: *Internet of Things; Blockchain; Smart Contract; bảo mật và toàn vẹn dữ liệu.*

I. GIỚI THIỆU

Internet vạn vật (IoT) là một trong những công nghệ chủ chốt của công nghiệp 4.0 và là mạng lưới của những thiết bị gia dụng, phương tiện giao thông, hoặc bất cứ thiết bị vật lý nào được tích hợp điện tử, phần mềm và kết nối. IoT giúp cho các thiết bị kết nối trở nên thông minh hơn. Chúng có thể được giám sát và điều khiển từ xa bởi con người, thậm chí chúng có thể giao tiếp, tương tác qua mạng Internet mà không cần có sự tương tác trực tiếp với con người.

IoT ra đời là nhờ sự phát triển của các công nghệ cảm biến, dữ liệu lớn, trí thông minh nhân tạo và cơ sở hạ tầng kết nối mạng. Trong hệ thống IoT, các cảm biến sẽ đóng vai trò là các giác quan giúp cho các thiết bị IoT thu thập dữ liệu về môi trường xung quanh. Dữ liệu thu thập được xử lý tại chỗ hoặc gửi lên máy chủ từ xa thông qua hạ tầng mạng để thực hiện những phân tích phức tạp. Dựa trên kết quả phân tích dữ liệu, những thiết bị IoT có thể thực hiện các hành động thông qua các thiết bị điều khiển cơ khí (actuator).

IoT ngày càng đóng vai trò quan trọng trong đời sống, được ứng dụng vô cùng rộng rãi trên hầu hết các lĩnh vực. Số lượng thiết bị IoT được Cisco dự đoán sẽ đạt 50 tỷ vào năm 2020 [1]. Chúng có thể là đèn thông minh, máy điều hòa thông minh trong nhà bạn, đến những thiết bị theo dõi sức khỏe như đồng hồ thông minh, máy đo đường huyết hay thậm chí là đèn giao thông thông minh trong thành phố. Nhờ vào những thiết bị IoT này, những khái niệm như nhà thông minh, công sở thông minh, nông trại, nhà máy hoặc thậm chí là thành phố thông minh đã không còn xa lạ. Hơn nữa, hàng loạt công nghệ truyền tải ra đời phục vụ cho mạng máy tính nói chung và IoT nói riêng như Bluetooth, WiFi, LTE, Zigbee, Z-Wave, 6LoWPAN, NFC, GSM, LoRa, NB-IoT và gần đây nhất là 5G.

Những thiết bị IoT thông minh, ứng dụng thông minh hoặc các dịch vụ thông minh như hiện nay chủ yếu được tạo ra trên cơ sở phân tích từ những dữ liệu thu thập dựa trên nền tảng IoT. Các dữ liệu này có thể phải được thu thập trong một thời gian đủ dài để tạo ra cơ sở tri thức cho các hệ thống phân tích dữ liệu. Và giải pháp thường được chọn để rút ngắn thời gian phát triển các ứng dụng IoT hiện nay là sử dụng dữ liệu chia sẻ từ những hệ thống thu thập dữ liệu hiện có. Tuy nhiên, việc quản lí chia sẻ dữ liệu không tốt có thể gây mất các thông tin nhạy cảm, ví dụ: tình trạng sức khỏe, mất quyền điều khiển thiết bị. Vì vậy, chúng ta cần có các cơ chế quản lí truy cập và chia sẻ dữ liệu thích hợp.

Các thiết bị đầu cuối trong các hệ thống IoT thường bị hạn chế về khả năng lưu trữ cũng như khả năng xử lí nên đa phần các cơ chế phức tạp như quản lý truy cập hay các giải thuật phức tạp trong quản lý dữ liệu chia sẻ thường được đảm nhận bởi các máy chủ tập trung. Tuy nhiên, mô hình tập trung lại có những nhược điểm lớn như tính sẵn sàng thấp, thiếu minh bạch, thiếu bảo mật, dễ bị thao túng từ bên trong và tấn công từ bên ngoài. Trong khi đó, sau sự thành công trong lĩnh vực tài chính, Blockchain đã chứng minh được tiềm năng ứng dụng trong nhiều lĩnh vực khác như giáo dục, y tế, nông nghiệp, quản lý chuỗi cung ứng và dịch vụ công nhờ các tính chất ưu việt của mình bao gồm bảo toàn dữ liệu, giao tiếp thông qua mạng ngang hàng dựa trên các luật đồng thuận, và tạo ra môi trường đảm bảo được tính minh bạch và toàn vẹn của dữ liệu.

Có thể thấy Blockchain có những tính chất có khả năng giải quyết các thách thức trong việc chia sẻ dữ liệu IoT như đã đề cập ở trên. Bên cạnh đó, các chuyên gia trong lĩnh vực nghiên cứu lẫn công nghiệp đều đang không ngừng nỗ lực chạy đua để kết hợp Blockchain và các hệ thống IoT hiện hữu để giải quyết những vấn đề khác nhau. Khả năng kết hợp giữa Blockchain và IoT trở nên rõ ràng hơn từ khi xuất hiện hợp đồng thông minh (smart contract), một chương trình máy tính tự động có thể thực thi các điều khoản của hợp đồng. Sau khi hợp đồng (contract) được biên dịch, nó sẽ được tải lên mạng Blockchain và được xác định bởi một địa chỉ duy nhất. Tất cả người dùng trong mạng

Blockchain có thể kích hoạt các chức năng trong contract bằng cách gửi transaction (giao dịch blockchain) đến contract. Nhờ vào môi trường Blockchain, việc thực thi của contract được đảm bảo chính xác và công minh.

Trong bài báo này, nhóm tác giả sẽ trình bày một mô hình sử dụng Blockchain kết hợp smart contract như là một giải pháp thay thế kiến trúc tập trung trong vấn đề quản lý truy cập và chia sẻ dữ liệu IoT. Phần còn lại của bài báo được trình bày như sau: một số nghiên cứu liên quan với đề tài của bài báo được giới thiệu ở phần II. Tiếp theo đó, phần III trình bày kiến trúc mô hình và mô tả hoạt động của hệ thống mà nhóm nghiên cứu đề xuất cho giải pháp quản lí chia sẻ dữ liệu IoT. Phần IV trình bày kết quả hiện thực một hệ thống thử nghiệm dựa trên mô hình đề xuất của nhóm. Phần V sẽ thảo luận những ưu nhược điểm của mô hình đề xuất dựa trên việc kết hợp và tận dụng các tính chất ưu việt của Blockchain. Và cuối cùng, nhóm nghiên cứu sẽ tóm tắt một số kết quả đã đạt được và đưa ra hướng phát triển tiếp theo.

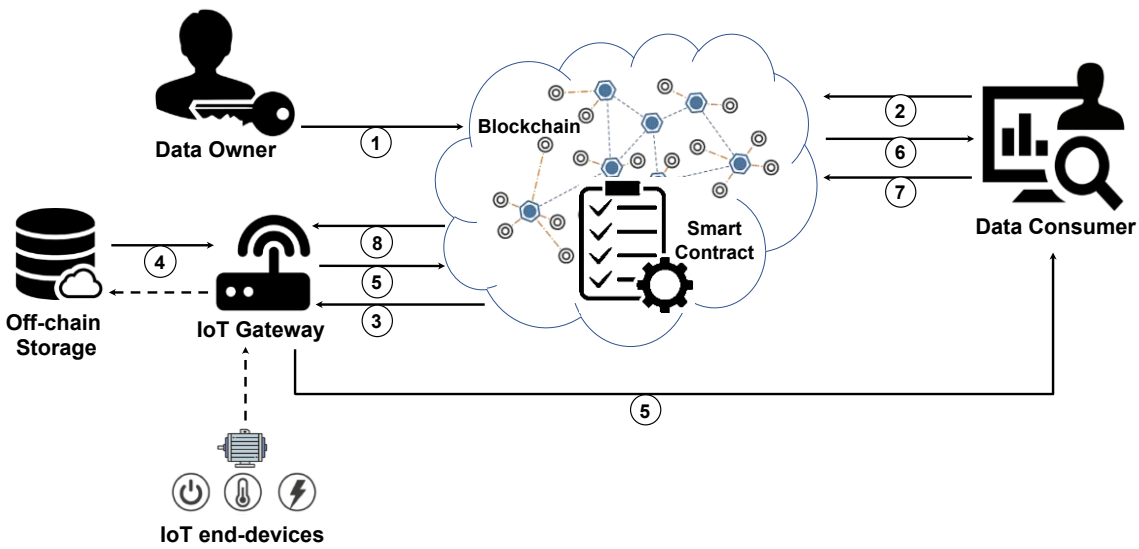
II. CÁC NGHIÊN CỨU LIÊN QUAN

Tương tự như chủ đề mà nhóm tác giả đang thực hiện, có rất nhiều nghiên cứu khác đã được thực hiện để chứng minh tiềm năng của việc kết hợp Blockchain vào IoT. Một số nghiên cứu tiêu biểu được tóm tắt như sau:

- Nhóm tác giả [3] đã thực hiện khảo sát hàng loạt mô hình phòng chống, phát hiện xâm nhập trong IoT, từ đó đưa ra 9 hướng nghiên cứu liên quan đến việc bảo mật hệ thống IoT sử dụng Blockchain. Trong khi đó, nhóm tác giả [4] đã thực hiện khảo sát nhiều phương án khác nhau và hệ thống thực tế kết hợp IoT và Blockchain.
- Nhóm tác giả [5] đề xuất một mạng Blockchain "lightweight" dùng cho hệ thống IoT với ví dụ cụ thể là hệ thống nhà thông minh. Nhóm tác giả này tiếp tục tiến hành các nghiên cứu đánh giá sâu hơn ở [6] và tổng hợp lại ở [7]. Nghiên cứu này cũng được áp dụng để phát triển mô hình cập nhật phần mềm đề xuất bởi nghiên cứu [8].
- IOTA [9] xây dựng một cấu trúc dữ liệu gọi là Tangle được sáng tạo từ Blockchain. Hệ thống này sử dụng đồng tiền mã hóa có tên IOTA, các giao dịch được thiết kế thích hợp cho việc trao đổi dữ liệu IoT.
- Nhóm tác giả [10] và [11] sử dụng smart contract hiện thực các cơ chế quản lí truy cập an toàn cho thiết bị IoT để đảm bảo tính toàn vẹn của dữ liệu IoT trong lưu trữ và trao đổi.
- Nghiên cứu [12] đưa ra bốn giao thức trên nền tảng Blockchain có sử dụng các hàm hash, hàm mã hóa, sử dụng trên cả dịch vụ lưu trữ đám mây.
- Nhờ re-encryption key, nghiên cứu [13], [14] sử dụng Blockchain node mà không cần tin tưởng như các proxy server (máy chủ trung gian) để chuyển tiếp dữ liệu giữa các bên một cách an toàn.

III. MÔ HÌNH ĐỀ XUẤT

Hình 1 mô tả kiến trúc của mô hình quản lý chia sẻ dữ liệu giữa Data Owner (những người sở hữu dữ liệu) và Data Consumer (những người muốn sử dụng dữ liệu) mà nhóm nghiên cứu đề xuất và hiện thực thử nghiệm. Data Owner có thể sở hữu nhiều thiết bị IoT sinh ra dữ liệu (IoT end-devices). Do các thiết bị IoT có khả năng xử lí thấp và sử dụng nhiều giao thức truyền dữ liệu khác nhau, chúng được kết nối đến Gateway như là cổng giao tiếp với môi trường Internet. Data Owner sử dụng Off-chain Storage (ví dụ: dịch vụ lưu trữ đám mây) để lưu trữ dữ liệu của mình.



Hình 1. Kiến trúc mô hình đề xuất

Có 02 quá trình bao gồm (1) quá trình thu thập dữ liệu và (2) quá trình quản lý chia sẻ dữ liệu bằng 08 bước được đánh dấu như trong Hình 1. Đối với quá trình thu thập dữ liệu (thể hiện bằng nét đứt trong hình).

- Các thiết bị IoT (IoT end-devices) được kết nối và gửi dữ liệu đến Gateway, dữ liệu được gửi đi bao gồm các dữ liệu mà thiết bị này thu thập được và thời gian thu thập dữ liệu đó, ngoài ra còn được đính kèm với mã ID của thiết bị để định danh trên Gateway.
- Sau khi nhận được dữ liệu từ các thiết bị IoT, Gateway sẽ tiến hành mã hóa dữ liệu này bằng public key của Data Owner sau đó lưu trữ lên **Off-chain Storage** theo giao thức được thiết lập sẵn. Với ID được gửi kèm theo dữ liệu từ thiết bị, Gateway sẽ lưu trữ dữ liệu này theo từng ID của từng thiết bị trên Off-chain Storage. Vì dữ liệu của các thiết bị IoT là dữ liệu theo thời gian nên Off-chain Storage được thiết lập theo cơ sở dữ liệu Realtime để có thể phù hợp với các thiết bị IoT.

Các bước trong quá trình quản lý chia sẻ dữ liệu được mô tả như sau

- **Bước 1**: Chủ sở hữu Data Owner sẽ đăng ký các thông tin về thiết bị của mình, xem danh sách thiết bị mà mình đã đăng ký, danh sách những người được chia sẻ dữ liệu thông qua Smart Contract.
- **Bước 2**: Data Consumer gửi vào Smart Contract một lượng tiền (chi phí cho việc sử dụng dữ liệu chia sẻ) tương ứng với thiết bị và khoảng thời gian họ muốn lấy dữ liệu, chi phí này sẽ được thông báo tại ứng dụng của Data Consumer. Giao dịch sẽ bị huỷ nếu Data Consumer không chuyển đủ chi phí cần thiết để nhận được dữ liệu chia sẻ.
- **Bước 3**: Gateway sẽ lắng nghe sự kiện yêu cầu truy cập dữ liệu thành công từ Smart Contract.
- **Bước 4**: Sau khi nhận được thông báo yêu cầu truy cập dữ liệu thành công từ Smart Contract, Gateway sẽ tiến hành trích xuất dữ liệu tương ứng với thời gian được yêu cầu từ Off-chain Storage về để giải mã, sau đó mã hóa bằng public key của Data Consumer, kèm theo đó Gateway sẽ hash dữ liệu sẽ được gửi đi và đính kèm mã hash này vào dữ liệu.
- **Bước 5**: Gateway sẽ tiến hành gửi dữ liệu thông qua API do Data Consumer cung cấp khi yêu cầu dữ liệu. Khi đã gửi dữ liệu đi, Gateway sẽ xác nhận trên Smart Contract là đã gửi dữ liệu.
- **Bước 6**: Data Consumer sẽ nhận được sự kiện từ Smart Contract thông báo dữ liệu yêu cầu đã được gửi, kèm theo đó là giá trị hash của dữ liệu.
- **Bước 7**: Data Consumer kiểm tra tính toàn vẹn của dữ liệu xác nhận đã nhận được dữ liệu đúng với dữ liệu được công bố với Smart Contract.
- **Bước 8**: Smart Contract gửi tiền do Data Consumer gửi vào Smart Contract lúc yêu cầu cho Data Owner.

IV. HIỆN THỰC

Để chứng minh tính khả thi cũng như đánh giá mô hình đề xuất, nhóm tác giả đã hiện thực thử nghiệm một hệ thống quản lý chia sẻ dữ liệu IoT dựa trên mô hình đã đề xuất trên Etherum - một nền tảng Blockchain mở và hỗ trợ triển khai các Smart Contract một cách dễ dàng. Trong mô hình đề xuất của nhóm, có 3 đối tượng chính là Data Owner, Data Consumer và các Smart Contract dùng để thực hiện các bước trong quá trình quản lý chia sẻ và trao đổi dữ liệu giữa Data Owner và Data Consummer. Như vậy, phần lõi chính của mô hình đề xuất là việc thiết kế và hiện thực Smart Contract.

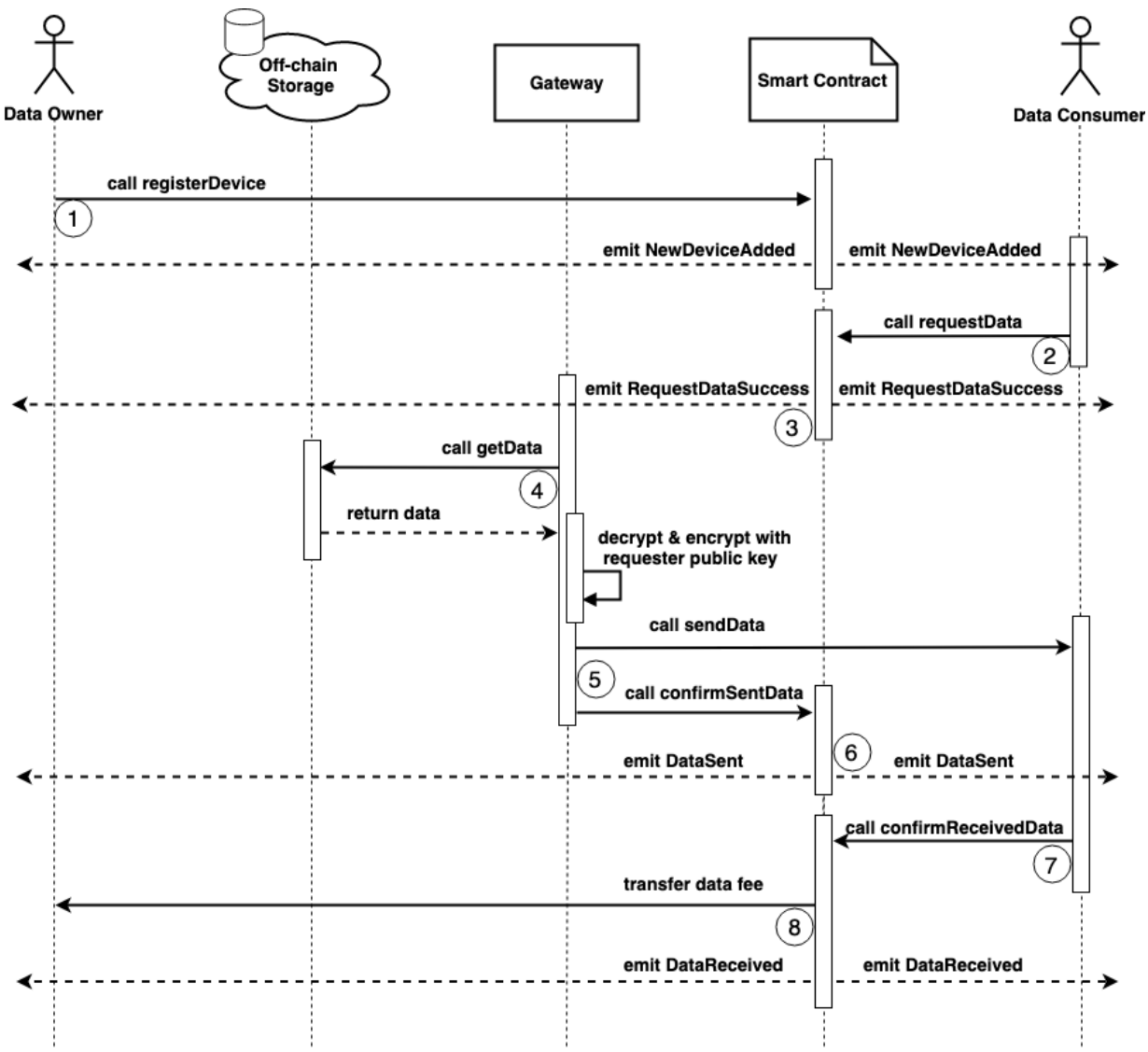
A. Smart Contract

Mỗi người dùng tham gia hệ thống được định danh bằng một chuỗi số duy nhất, cũng là địa chỉ dùng trong Ethereum. Smart contract lưu trữ danh sách những Data Owner, mỗi người sẽ sở hữu một danh sách những thiết bị mà mình đã đăng ký trên hệ thống. Khi đăng ký thiết bị, Data Owner phải cung cấp hai thông tin là định danh thiết bị deviceId và giá tiền theo ngày dailyPrice mà người dùng phải bỏ ra để mua dữ liệu của thiết bị đó.

Do quá trình thực thi smart contract xảy ra trong môi trường máy ảo Ethereum (EVM), smart contract không thể thông báo cho môi trường bên ngoài biết các sự kiện đang diễn ra bên trong. Để làm được điều này, smart contract hỗ trợ tạo ra các event có chứa các thông tin nhất định và người dùng bên ngoài môi trường Blockchain có thể lắng nghe các event đó. Hình 2 mô tả hiện thực bằng lược đồ tuần tự của quá trình quản lý và trao đổi dữ liệu giữa Data Owner và Data Consumer dựa trên các sự kiện và thực thi các Smart Contract tương ứng với 08 bước như trong kiến trúc mô hình đề xuất. Cụ thể

- **Bước 1**: Khi thiết bị được đăng ký thành công, một event **NewDeviceAdded** được phát ra và Data Consumer có thể biết được sự kiện trên.
- **Bước 2**: Để yêu cầu dữ liệu, Data Consumer chọn đúng định danh thiết bị, dữ liệu cần mua trong khoảng thời gian nào, số tiền cần thiết và gọi hàm ***requestData*.** Đồng thời, để Gateway có thể mã hóa dữ liệu và gửi đến đúng địa chỉ, Data Consumer cũng cung cấp public key publicKey của mình và API mà mình muốn Gateway gửi dữ liệu đến.

- **Bước 3**: Khi nhận được yêu cầu từ Data Consumer, Smart Contract kiểm tra các điều kiện yêu cầu dữ liệu có hợp lý không, thiết bị được yêu cầu có tồn tại không, số tiền Data Consumer gửi đến có đủ không. Nếu yêu cầu dữ liệu không hợp lệ, một event **RequestDataFail** được phát ra. Ngược lại, trong trường hợp yêu cầu hợp lệ, một event **RequestDataSuccess** được phát ra, Smart Contract sẽ tính giá trị hash từ ba thông số là định danh của Data Owner, định danh của Data Consumer và thời gian của yêu cầu, giá trị hash này được xem như là định danh của giao dịch.
- **Bước 4**: Khi Gateway lắng nghe được event **RequestDataSuccess**, Gateway sẽ sử dụng các thông tin cần thiết từ event, lấy dữ liệu tương ứng được yêu cầu từ Off-chain Storage bằng cách gọi hàm ***getData***, thực hiện các quá trình mã hóa dữ liệu bằng public-private key của Data Owner. Sau đó, dữ liệu sẽ được mã hoá bằng public key của Data Consumer.
- **Bước 5**: Gateway gửi dữ liệu đến API mà Data Consumer cung cấp bằng cách gọi hàm ***sendData***. Đồng thời, Gateway cũng gọi hàm ***confirmSentData*** của Smart Contract, trong đó có gửi kèm định danh của giao dịch và giá trị hash của dữ liệu.
- **Bước 6**: Một khi Smart Contract nhận được xác nhận đã gửi thông tin của Data Owner, Smart Contract phát ra event **DataSent**.
- **Bước 7**: Khi Data Consumer nhận được sự kiện **DataSent** là biết mình đã nhận được dữ liệu. Data Consumer sẽ tiến hành giải mã dữ liệu, tính giá trị hash của dữ liệu nhận được và so sánh với giá trị hash được Data Owner cung cấp. Nếu trùng khớp, nghĩa là dữ liệu đảm bảo được tính toàn vẹn, Data Consumer gọi cách hàm ***confirmReceivedData*** để xác nhận là mình đã nhận dữ liệu thành công.
- **Bước 8**: Smart Contract gửi số tiền mà Data Consumer đã đặt trước đó đến cho Data Owner và phát ra event **DataReceived** để Data Owner biết được rằng Data Consumer đã nhận được dữ liệu thành công.



Hình 2. Lược đồ tuần tự quá trình quản lý chia sẻ và trao đổi dữ liệu trong mô hình đề xuất

B. Data Owner

Một Data Owner sẽ quản lý 03 thành phần bao gồm (1) Off-chain Storage, (2) Gateway và (3) Thiết bị IoT (IoT end-devices) như trong mô hình đề xuất. Các thành phần được hiện thực và mô tả như sau.

1. **Off-chain Storage**: có chức năng lưu trữ dữ liệu IoT thu thập được từ các thiết bị IoT, ví dụ: nhiệt độ, độ ẩm. Các dữ liệu này được lưu trữ theo định dạng sau:

```
{
    times: {...}, // mốc thời gian dữ liệu được thu thập
    value: {...}  // giá trị dữ liệu
}
```

Trong quá trình hiện thực hệ thống, nhóm tác giả sử dụng cơ sở dữ liệu RealTime FireBase do Google cung cấp đóng vai trò là Off-chain Storage trong mô hình đề xuất. FireBase là một nền tảng ứng dụng di động và web với các công cụ và hạ tầng được thiết kế để giúp các lập trình viên xây dựng các ứng dụng một cách nhanh chóng và chất lượng nhất.

2. **Gateway**: được hiện thực bằng Nodejs chạy trên board Raspberry Pi3 và có giao diện được hiện thực bằng HTML và JavaScript. Gateway sẽ bao gồm các chức năng chính sau đây:

- Nhận dữ liệu từ các thiết bị IoT và mã hóa các dữ liệu này sau đó lưu trữ dữ liệu lên Firebase.
- Lấy dữ liệu từ Firebase về để giải mã, sau đó lọc ra các dữ liệu cần thiết, tiếp theo mã hóa lại bằng public key của người yêu cầu dữ liệu và gửi dữ liệu này đến cho họ theo định dạng:

```
{
        Hash: {..}          // giá trị hash của dữ liệu để kiểm tra tính đúng đắn và toàn vẹn
        RequestData: {..} // dữ liệu IoT được yêu cầu
}
```

- Hỗ trợ giao diện người dùng để cho chủ sở hữu dữ liệu tương tác với Smart Contract: đăng ký thiết bị mới, lấy danh sách các thiết bị đã đăng ký và danh sách những người yêu cầu dữ liệu, cũng như thống kê các dữ liệu được thu thập từ các thiết bị IoT. Ngoài ra còn hỗ trợ tính năng đăng nhập để tăng tính bảo mật của hệ thống.

3. **Thiết bị IoT**: bao gồm các thiết bị có chức năng thu thập dữ liệu và gửi đến Gateway để lưu trữ và dữ liệu này được gửi đi một cách liên tục. Và các thiết bị trong bài báo này được hiện thực bằng ngôn ngữ C trên phần cứng nhúng arduino. Các thư viện được sử dụng trong phần hiện thực bao gồm:

- Web3.js dùng để tương tác với Smart Contract thông qua Front-end của Gateway.
- Eth-Crypto: dùng để giải mã và mã hóa dữ liệu bằng Public Key và Private key.
- FireBase: dùng để liên kết và sử dụng cơ sở dữ liệu của FireBase.
- Johnny-Five: dùng để giao tiếp giữa các thiết bị IoT (Arduino) với Server Nodejs

C. Data Consumer

Các chức năng của Data Consumer được hiện thực bằng ngôn ngữ JavaScript theo mô hình Mern Stack. Người dùng tương tác với hệ thống thông qua giao diện web được viết bằng Reactjs cùng với phần Backend được hiện thực bằng Nodejs, Express và MongoDB. Cụ thể, một Data Consumer có các chức năng sau:

- Tạo ví Ethereum để thực hiện thanh toán cho Data Owner khi yêu cầu chia sẻ dữ liệu
- Hiển thị danh sách các Data Owner và thông tin về các thiết bị IoT mà họ sở hữu.
- Thực hiện yêu cầu dữ liệu từ các thiết bị IoT
- Xác nhận nhận được dữ liệu đã yêu cầu từ Data Owner
- Giải mã dữ liệu đã mã hóa nhận được từ Data Owner.

Trong quá trình hiện thực ứng dụng cho Data Consumer, nhóm tác giả sử dụng các thư viện sau đây:

- Ethereumjs-wallet và Ethereumjs-util để tạo ví Ethereum
- Web3.js và Ethereumjs-tx dùng để ký và gửi transaction đến mạng Ethereum
- Eth-Crypto: dùng để giải mã dữ liệu bằng Private Key của tài khoản Ethereum

V. ĐÁNH GIÁ HỆ THỐNG

Trong mô hình hệ thống đề xuất, việc quản lý chia sẻ và trao đổi dữ liệu được thực hiện thông qua smart contract – đây có thể xem là phần lõi của hệ thống đề xuất. Do đó, nhóm đã thực hiện kiểm tra tính đúng đắn của smart

contract bằng cách sử dụng Truffle tạo môi trường Ethereum ảo để thực hiện unit test cho smart contract. 100 tài khoản được khởi tạo và mỗi tài khoản sẽ đóng vài trò là Data Consumer và Data Owner để thực hiện các chức năng tương tác với smart contract trong quá trình trao đổi dữ liệu. Kết quả cho thấy 100% các chức năng hoạt động của smart contract đúng với thiết kế.

Mô hình đề xuất của nhóm tác giả kết hợp Blockchain và Smart Contract nên hệ thống đã tận dụng được những lợi thế như: Smart contract thay thế bên thứ ba trung gian giao dịch, từ đó giúp giảm chi phí. Hơn nữa, nền tảng phi tập trung của Blockchain giúp tăng tính sẵn sàng và hạn chế tấn công DoS cho dịch vụ mà Smart Contract cung cấp; Tận dụng được hệ thống private key, public key sẵn có của blockchain để mã hóa dữ liệu, đảm bảo tính bảo mật cho dữ liệu; và toàn bộ quá trình giao dịch được truy vết. Từ đó, hệ thống mà nhóm đề xuất đáp ứng được 3 tiêu chí cơ bản về bảo mật dữ liệu như sau.

- Tính bảo mật, riêng tư (Confidentiality): Cơ chế cấp quyền và mã hóa được áp dụng, đảm bảo chỉ Data Owner và những người được cấp quyền mới được truy cập và đọc được dữ liệu.
- Tính toàn vẹn dữ liệu (Integrity): Trong quá trình trao đổi dữ liệu, giá trị hash được sử dụng làm bằng chứng cho việc dữ liệu không bị thay đổi. Ngoài ra, Blockchain đảm bảo các giao dịch xảy ra là không thể xóa sửa.
- Tính sẵn sàng (Availability): Nhờ thay thế mô hình tập trung, dịch vụ của Smart Contract mang lại tính sẵn sàng cao. Dịch vụ xử lý của Gateway cũng được đảm bảo hoạt động tốt nhờ cơ chế cấp quyền và cơ chế chống yêu cầu liên tục của Smart Contract.

Hơn nữa, về tốc độ xử lý so với giao dịch ở thế giới thực, hệ thống tránh được sự tranh chấp và tăng tốc độ giao dịch. Tuy nhiên, tốc độ xử lí giao dịch sẽ bị hạn chế bởi tốc độ xử lý của mạng Blockchain tuỳ vào thời điểm sẽ có thể khác nhau (ví dụ: về mặt lý thuyết: mất khoảng 12giây để đóng một block trong mạng Ethereum). Do đó, thực tế hệ thống sẽ hoạt động chậm nếu số lượng giao dịch bùng nổ. Nhóm tác giả cũng đã tiến hành thử nghiệm đo đạc thời gian tính từ lúc Data Consumer gởi yêu cầu dữ liệu cho đến khi Data Consumer nhận được dữ liệu của Data Owner theo mô hình đề xuất. Quá trình thực nghiệm được thực hiện trong 50 lần với những thời điểm khác nhau và được tổng hợp trong Bảng 1.

Bảng 1. Kết quả thực nghiệm về thời gian xử lý một quá trình trao đổi dữ liệu giữa Data Consumer và Data Owner

Thời gian tối thiểu	Thời gian trung bình	Thời gian tối đa	Độ lệch chuẩn
67 (giây)	111 (giây)	165 (giây)	26.8 (giây)

VI. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Khi số lượng thiết bị IoT ngày càng tăng lên, việc kiểm soát truy cập dữ liệu và minh bạch sử dụng dữ liệu rất quan trọng do lượng dữ liệu khổng lồ được tạo ra từ các thiết bị này. Trong bài báo này, nhóm tác giả đã trình bày mô hình hiện thực cho việc quản lý chia sẻ dữ liệu phi tập trung trong IoT nhằm nâng cao tính bảo mật, riêng tư và toàn vẹn dữ liệu dựa trên việc tận dụng những tính chất ưu việt của công nghệ Blockchain kết hợp smart contract. Kết quả hiện thực thử nghiệm cho thấy tiềm năng và tính khả thi của giải pháp đề xuất.

Trong thời gian tới nhóm nghiên cứu sẽ tiếp bổ sung các chức năng về quản lý quyền truy cập khác nhau đối với thiết bị IoT, cụ thể là quyền truy cập (authorization và authentication) đối với các thiết bị IoT và hệ thống để tăng cường tính bảo mật trên thiết bị và hệ thống.

VII. LỜI CẢM ƠN

Nhóm tác giả trân trọng cảm ơn sự hỗ trợ và tư vấn kỹ thuật của các chuyên gia của công ty Infinity Blockchain Labs, Công ty Vietnam Blockchain Corporation và Công ty VieTech Holdings trong quá trình thực hiện nghiên cứu này.

TÀI LIỆU THAM KHẢO

[1] Dave Evans, “The Internet of Things: How the next evolution of the internet is changing everything”, Cisco White Paper. [Online] Available at: https://bit.ly/2b6Wm55

[2] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system”, Bitcoin White Paper, 2018. [Online] Available at: https://bitcoin.org/bitcoin.pdf

[3] M. Banerjee, J. Lee, and K.-K. R. Choo, “A blockchain future for internet of things security: a position paper”, *Digital Communications and Networks*, vol. 4 (3), pp. 149–160, 2018.

[4] E. F. Jesus, V. R. L. Chicarino, C. V. N. Albuquerque, and A. A.de A. Rocha, “A Survey of How to Use Blockchain to Secure Internet of Things and the Stalker Attack”, *Security and Communication Networks*, vol. 2018, pp. 1-27, 2018.

[5] A. Dorri, S. S. Kanhere, and R. Jurdak: "Blockchain in Internet of Things: Challenges and Solutions", *arXiv preprint*, vol. 1608.05187, 2016.

[6] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home", *in Proceedings of 2017 IEEE Conference on Pervasive Computing and Communications*, pp. 618-623, 2017.

[7] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for IoT", *in Proceedings of the 2nd International Conference on Internet-of-Things Design and Implementation*, pp. 173-178, 2017.

[8] M. Steger, A. Dorri, S. S. Kanhere, K. R¨omer, R. Jurdak, and M. Karner, "Secure Wireless Automotive Software Updates Using Blockchains: A Proof of Concept", *Advanced Microsystems for Automotive Applications*, Lecture Notes in Mobility, pp. 137-149, 2017.

[9] S. Popov, "The tangle", IOTA White Paper, 2016. [Online] Available at: https://www.iota.org/research/academic-papers

[10] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart Contract-Based Access Control for the Internet of Things", *IEEE Internet of Things Journal*, vol 6 (2), pp. 1594 - 1605, 2019.

[11] K. Salah, "A User Authentication Scheme of IoT Devices using Blockchain-enabled Fog Nodes", *in Proceedings of 15th ACS/IEEE International Conference on Computer Systems and Applications*, pp. 1-8, 2018.

[12] B. Liu, X. L. Yu, S. Chen, X. Xu, L. Zhu, "Blockchain based data integrity service framework for IoT data", *in Proceedings of IEEE International Conference on Web Services*, pp. 468-475, 2017.

[13] Truc D. T. Nguyen, Hoang-Anh Pham, My T. Thai, "Leveraging Blockchain to Enhance Data Privacy in IoT-Based Applications", *in Proceedings of the 7th International Conference on Computational Data and Social Networks*, pp. 211-221, 2018.

[14] O. Agyekum, Q. Xia, E. B. Sifah, J. Gao, H. Xia, X. Du, and M. Guizani, "Secured Proxy-Based Data Sharing Module in IoT Environments Using Blockchain", *Sensors 2019*, vol 19 (5), 1235. [Online] Available at: https://doi.org/10.3390/s19051235

ENHANCE THE INTEGRITY AND SECURITY IN IOT DATA STORAGE AND SHARING MANAGEMENT BY USING BLOCKCHAIN

Le Trung Kien, Pham Thi Ngoc My, Nguyen Hoai Quoc Trung, Pham Hoang Anh*

ABSTRACT: *The needs of sharing available data to shorten time-consuming and cost savings in development and deployment of smart systems based IoT platforms are more practical. However, there are many challenges in ensuring integrity, security, and fairness in the data sharing process. Additionally, IoT devices for data collection are often limited by processing capability to perform complicated tasks, and they are vulnerable due to the centralized model like most conventional IoT systems. Meanwhile, the emerging Blockchain technology has shown its potential to help in solving these challenges, as mentioned above. In this paper, the authors propose a solution leveraging Blockchain to enhance the integrity and security of IoT data storage and sharing management. Our prototype demonstrates the feasibility of the proposed solution.*