(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)**
CỤC SỞ HỮU TRÍ TUỆ

(11) **1-0044818**

(51)[2020.01] **G06F 9/445**; H04L 29/08

(13) **B**

(21) 1-2021-03107
(22) 09/11/2018
(86) PCT/CN2018/114911 09/11/2018
(87) WO2020/093398 14/05/2020
(45) 25/04/2025 445
(43) 25/08/2021 401A

(71) HUAWEI TECHNOLOGIES CO., LTD. (CN)
Huawei Administration Building, Bantian, Longgang District, Shenzhen, Guangdong 518129, P. R. China
(72) LI, Shubin (CN).
(74) Công ty Luật TNHH T&G (TGVN)

(54) PHƯƠNG PHÁP NÂNG CẤP QUA VÔ TUYẾN, PHƯƠNG PHÁP GỬI DỮ LIỆU BẰNG CÔNG NGHỆ QUA VÔ TUYẾN, MÁY CHỦ VÀ THIẾT BỊ ĐẦU CUỐI DI ĐỘNG

(21) 1-2021-03107

(57) Sáng chế đề xuất phương pháp nâng cấp qua vô tuyến, bao gồm: trước tiên, máy chủ thứ nhất thu được phiên bản mới của dữ liệu được mã hóa và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống được áp dụng cho thiết bị đầu cuối di động; sau đó, máy chủ thứ nhất giải mã phiên bản mới của dữ liệu được mã hóa để thu được phiên bản mới của dữ liệu ban đầu, và giải mã phiên bản cũ của dữ liệu được mã hóa để thu được phiên bản cũ của dữ liệu ban đầu; tiếp theo, máy chủ thứ nhất thực hiện phép sai phân trên phiên bản mới của dữ liệu ban đầu và phiên bản cũ của dữ liệu ban đầu để thu được dữ liệu sai phân, và tạo ra dữ liệu OTA dựa trên dữ liệu sai phân; máy chủ thứ nhất chuyển dữ liệu OTA tới thiết bị đầu cuối di động, và thiết bị đầu cuối di động giải mã dữ liệu OTA để thu được dữ liệu sai phân, và sau đó hồi phục phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và thu được phiên bản cũ của dữ liệu ban đầu; và cuối cùng, thiết bị đầu cuối di động nâng cấp phiên bản cũ của phần mềm hệ thống cho phiên bản mới bằng cách sử dụng phiên bản mới của dữ liệu ban đầu. Theo cách này, kích thước của gói nâng cấp OTA có thể được giảm, và việc nâng cấp nhanh của phần mềm hệ thống trên thiết bị đầu cuối di động được thực hiện. Sáng chế cũng đề cập đến phương pháp gửi dữ liệu bằng công nghệ qua vô tuyến, máy chủ và thiết bị đầu cuối di động.

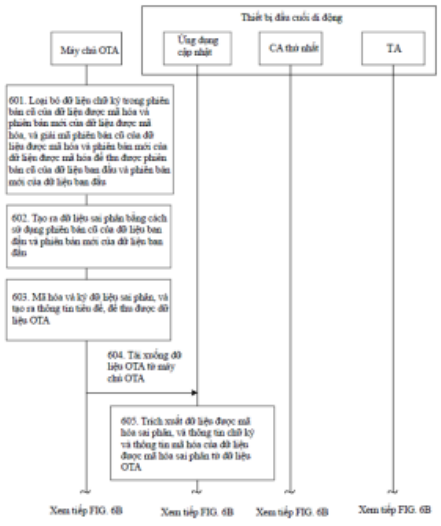


FIG. 6a

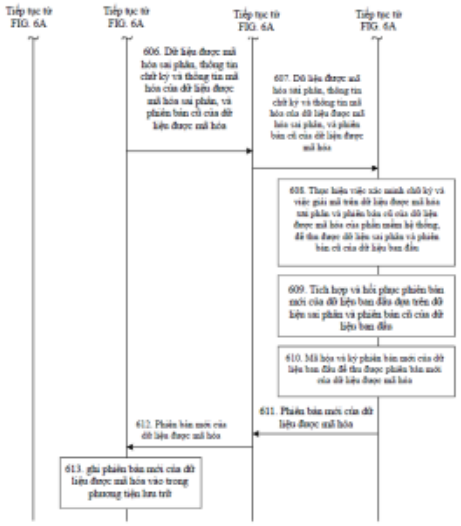


FIG. 6b

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập tới lĩnh vực công nghệ truyền thông, và cụ thể là, đề cập đến phương pháp nâng cấp qua vô tuyến và thiết bị liên quan.

Tình trạng kỹ thuật của sáng chế

Người dùng của các thiết bị đầu cuối di động tăng nhanh chóng, và các yêu cầu để cải thiện công nghệ truyền thông và các thiết bị đầu cuối di động cũng tăng cao. Các thiết bị đầu cuối di động cũng thông minh hơn và dựa trên đa phương tiện, và thường được cài đặt với phần mềm hệ thống khác nhau. Mặc dù các thiết bị đầu cuối di động này có các chức năng phong phú, phần mềm hệ thống cũng dễ tạo ra các cuộc tấn công hơn. Do đó, phần mềm ứng dụng của các thiết bị đầu cuối di động cần được nâng cấp và được cập nhật.

Hiện tại, công nghệ qua vô tuyến (over-the-air technology, OTA) được áp dụng cho việc nâng cấp và cập nhật phần mềm hệ thống của thiết bị đầu cuối di động do phần mềm hệ thống của thiết bị đầu cuối di động có thể được quản lý từ xa thông qua giao diện không gian của máy chủ. Cách nâng cấp OTA đang chủ yếu thực hiện nâng cấp ở chế độ máy khách/máy chủ. Thiết bị đầu cuối di động có thể tải xuống gói nâng cấp OTA từ máy chủ để nâng cấp phần mềm hệ thống trên thiết bị đầu cuối di động. Nếu lượng dữ liệu của gói nâng cấp OTA là quá lớn, khi Wi-Fi không được tạo cấu hình trên thiết bị đầu cuối di động, vấn đề mà người dùng lo lắng là vấn đề lưu lượng. Ngoài ra, khi lượng dữ liệu của gói nâng cấp OTA là quá lớn, các vấn đề như gói nâng cấp OTA không hoàn thành được tải xuống bởi thiết bị đầu cuối di động và lỗi nâng cấp cập nhật có thể xảy ra.

Do đó, cách để làm giảm kích thước của gói nâng cấp OTA trở thành vấn đề kỹ

thuật cần được giải quyết gấp bởi người có hiểu biết trung bình trong lĩnh vực tương ứng.

Bản chất kỹ thuật của sáng chế

Sáng chế đề xuất phương pháp nâng cấp bằng công nghệ qua vô tuyến và thiết bị liên quan, để làm giảm kích thước của gói nâng cấp OTA, và thực hiện việc nâng cấp nhanh phần mềm hệ thống trên thiết bị đầu cuối di động.

Theo khía cạnh thứ nhất, sáng chế đề xuất phương pháp gửi dữ liệu bằng công nghệ qua vô tuyến OTA, bao gồm các bước: Trước tiên, máy chủ thứ nhất thu được phiên bản mới của dữ liệu được mã hóa và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống được áp dụng cho thiết bị đầu cuối di động. Sau đó, máy chủ thứ nhất giải mã phiên bản mới của dữ liệu được mã hóa để thu được phiên bản mới của dữ liệu ban đầu, và giải mã phiên bản cũ của dữ liệu được mã hóa để thu được phiên bản cũ của dữ liệu ban đầu. Tiếp theo, máy chủ thứ nhất thực hiện phép sai phân trên phiên bản mới của dữ liệu ban đầu và phiên bản cũ của dữ liệu ban đầu, để thu được dữ liệu sai phân. Dữ liệu sai phân bao gồm dữ liệu mà nằm trong phiên bản mới của dữ liệu ban đầu và khác với phiên bản cũ của dữ liệu ban đầu. Tiếp theo, máy chủ thứ nhất tạo ra dữ liệu bằng công nghệ qua vô tuyến OTA. Dữ liệu OTA bao gồm dữ liệu được mã hóa sai phân và thông tin tiêu đề, và dữ liệu được mã hóa sai phân là dữ liệu thu được sau khi dữ liệu sai phân được mã hóa, hoặc dữ liệu được mã hóa sai phân là dữ liệu thu được sau khi dữ liệu sai phân được mã hóa và ký. Thông tin tiêu đề bao gồm thông tin tiêu đề mã hóa dữ liệu sai phân, hoặc thông tin tiêu đề bao gồm thông tin tiêu đề mã hóa dữ liệu sai phân và thông tin tiêu đề mã hóa dữ liệu đích. Thông tin tiêu đề mã hóa dữ liệu sai phân được sử dụng bởi thiết bị đầu cuối di động để giải mã dữ liệu được mã hóa sai phân để thu được dữ liệu sai phân, và thông tin tiêu đề mã hóa dữ liệu đích được sử dụng bởi thiết bị đầu cuối di động để xác minh chữ ký của dữ liệu được mã hóa sai phân. Cuối cùng, máy chủ thứ nhất gửi dữ liệu OTA tới thiết bị đầu cuối di động. Dữ liệu OTA được sử dụng để nâng cấp phiên bản cũ (ví dụ, phiên bản 8.1 của hệ điều hành EMUI) của phần mềm hệ thống của thiết bị đầu cuối di động thành phiên bản mới (ví dụ, phiên bản 8.2 của hệ điều hành EMUI).

Máy chủ thứ nhất theo phương án này của sáng chế là máy chủ OTA. Theo phương án thực hiện này của sáng chế, khi tạo ra dữ liệu OTA, máy chủ thứ nhất (cụ thể là, máy chủ OTA) có thể tạo ra dữ liệu sai phân dựa trên phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu mà thu được thông qua hoạt động giải mã, và gửi dữ liệu OTA bao gồm dữ liệu sai phân tới thiết bị đầu cuối di động. Do máy chủ OTA thực hiện phép sai phân chỉ trên phiên bản mới của dữ liệu ban đầu và phiên bản cũ của dữ liệu ban đầu, dữ liệu chữ ký của phiên bản mới của dữ liệu ban đầu được loại bỏ khỏi dữ liệu sai phân, sao cho kích thước của dữ liệu được truyền bởi máy chủ OTA tới thiết bị đầu cuối di động cho việc nâng cấp phần mềm hệ thống được giảm, và việc nâng cấp nhanh của phần mềm hệ thống được thực hiện.

Theo một phương án, việc máy chủ thứ nhất tạo ra dữ liệu OTA bao gồm: Trước tiên, máy chủ thứ nhất mã hóa dữ liệu sai phân để thu được dữ liệu được mã hóa sai phân, hoặc mã hóa và ký dữ liệu sai phân để thu được dữ liệu được mã hóa sai phân. Sau đó, máy chủ thứ nhất tạo ra thông tin tiêu đề cho dữ liệu được mã hóa sai phân. Tiếp theo, máy chủ thứ nhất tích hợp dữ liệu được mã hóa sai phân và thông tin tiêu đề để thu được dữ liệu OTA.

Theo một phương án, trước khi máy chủ thứ nhất thu được phiên bản mới của dữ liệu được mã hóa và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống được áp dụng cho thiết bị đầu cuối di động, phương pháp này còn bao gồm các bước: Trước tiên, máy chủ thứ nhất nhận phiên bản cũ của thông tin mà thuộc về phần mềm hệ thống và được gửi bởi thiết bị đầu cuối di động; và sau đó, máy chủ thứ nhất xác định phiên bản cũ của dữ liệu được mã hóa từ một hoặc nhiều phiên bản lịch sử của dữ liệu được mã hóa của phần mềm hệ thống dựa trên phiên bản cũ của thông tin của phần mềm hệ thống. Do máy chủ thứ nhất có thể có nhiều phiên bản lịch sử của dữ liệu của phần mềm hệ thống, bằng cách nhận phiên bản cũ của thông tin được gửi bởi thiết bị đầu cuối di động, máy chủ thứ nhất có thể xác định phiên bản cũ của dữ liệu được mã hóa từ nhiều phiên bản lịch sử của dữ liệu, và tạo ra dữ liệu sai phân trên phiên bản cũ, để đảm bảo sự chính xác của dữ liệu sai phân dùng cho việc nâng cấp phần mềm hệ thống trên thiết bị đầu cuối di động.

Theo một phương án, việc máy chủ thứ nhất giải mã phiên bản mới của dữ liệu được mã hóa để thu được phiên bản mới của dữ liệu ban đầu, và giải mã phiên bản cũ

của dữ liệu được mã hóa để thu được phiên bản cũ của dữ liệu ban đầu bao gồm: Trước tiên, máy chủ thứ nhất loại bỏ dữ liệu chữ ký ra khỏi phiên bản mới của dữ liệu được mã hóa, để thu được phiên bản mới của dữ liệu ban đầu được mã hóa; máy chủ thứ nhất loại bỏ dữ liệu chữ ký ra khỏi phiên bản cũ của dữ liệu được mã hóa, để thu được phiên bản cũ của dữ liệu ban đầu được mã hóa; và máy chủ thứ nhất giải mã phiên bản mới của dữ liệu ban đầu được mã hóa để thu được phiên bản mới của dữ liệu ban đầu, và giải mã phiên bản cũ của dữ liệu ban đầu được mã hóa để thu được phiên bản cũ của dữ liệu ban đầu. Dữ liệu chữ ký trong phiên bản mới của dữ liệu được mã hóa tăng khi phiên bản mới của dữ liệu ban đầu tăng. Theo cách này, sau khi dữ liệu chữ ký được loại bỏ, và lượng dữ liệu của dữ liệu sai phân được giảm, sao cho kích thước của dữ liệu được truyền bởi máy chủ OTA tới thiết bị đầu cuối di động cho việc nâng cấp phần mềm hệ thống được giảm, và việc nâng cấp nhanh của phần mềm hệ thống được thực hiện.

Theo một phương án, phiên bản cũ của dữ liệu ban đầu bao gồm phiên bản cũ của khối dữ liệu tệp, phiên bản cũ của khối dữ liệu hàm băm, và phiên bản cũ của khối dữ liệu kiểm tra. Phiên bản mới của dữ liệu ban đầu bao gồm phiên bản mới của khối dữ liệu tệp, phiên bản mới của khối dữ liệu hàm băm, và phiên bản mới của khối dữ liệu kiểm tra. Phiên bản cũ của khối dữ liệu kiểm tra là dữ liệu chữ ký của giá trị hàm băm gốc trong phiên bản cũ của khối dữ liệu hàm băm, và phiên bản cũ của khối dữ liệu hàm băm được sử dụng để kiểm tra phiên bản cũ của khối dữ liệu tệp; Phiên bản mới của khối dữ liệu kiểm tra là dữ liệu chữ ký của giá trị hàm băm gốc trong phiên bản mới của khối dữ liệu hàm băm, và phiên bản mới của khối dữ liệu hàm băm được sử dụng để kiểm tra phiên bản mới của khối dữ liệu tệp. Việc máy chủ thứ nhất thực hiện phép sai phân trên phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu, để thu được dữ liệu sai phân bao gồm: Trước tiên, máy chủ thứ nhất loại bỏ phiên bản cũ của khối dữ liệu hàm băm khỏi phiên bản cũ của dữ liệu ban đầu, và loại bỏ phiên bản mới của khối dữ liệu hàm băm khỏi phiên bản mới của dữ liệu ban đầu. Sau đó, máy chủ thứ nhất thực hiện phép sai phân trên khối dữ liệu dựa trên phiên bản cũ của dữ liệu ban đầu mà phiên bản cũ của khối dữ liệu hàm băm được loại bỏ khỏi đó và phiên bản mới của dữ liệu ban đầu mà phiên bản mới của khối dữ liệu hàm băm được loại bỏ khỏi đó, để thu được dữ liệu sai phân. Theo cách này, khi tạo ra dữ liệu

sai phân, máy chủ thứ nhất (máy chủ OTA) loại bỏ các khối dữ liệu tương ứng với cây hàm băm, và giữ lại phiên bản mới của khối dữ liệu kiểm tra và phiên bản mới của khối dữ liệu tệp, và sau đó khi thiết bị đầu cuối di động hồi phục dữ liệu sai phân, bằng cách sử dụng giá trị ngẫu nhiên kiểm tra trong phiên bản mới được hồi phục của khối dữ liệu tệp và phiên bản mới của khối dữ liệu kiểm tra, khối dữ liệu hàm băm tương ứng với phiên bản mới của khối dữ liệu tệp được tạo ra để được lưu trữ trong phương tiện lưu trữ. Do phiên bản mới của khối dữ liệu hàm băm được loại bỏ khỏi dữ liệu OTA, và phiên bản mới của khối dữ liệu hàm băm được tạo ra trên thiết bị đầu cuối di động được sử dụng để làm giảm lượng dữ liệu của dữ liệu cập nhật phần mềm hệ thống (dữ liệu OTA) được phát hành bởi máy chủ thứ nhất.

Theo khía cạnh thứ hai, sáng chế đề xuất phương pháp nâng cấp qua vô tuyến, bao gồm: Trước tiên, thiết bị đầu cuối di động nhận dữ liệu OTA được gửi bởi máy chủ thứ nhất. Dữ liệu OTA được sử dụng để nâng cấp phiên bản cũ (ví dụ, phiên bản 8.1 của hệ điều hành EMUI) của phần mềm hệ thống (ví dụ, hệ điều hành EMUI) trên thiết bị đầu cuối di động thành phiên bản mới (ví dụ, phiên bản 8.2 của hệ điều hành EMUI). Dữ liệu OTA bao gồm thông tin tiêu đề và dữ liệu được mã hóa sai phân, và thông tin tiêu đề bao gồm thông tin tiêu đề mã hóa dữ liệu sai phân, hoặc thông tin tiêu đề bao gồm thông tin tiêu đề mã hóa dữ liệu sai phân và thông tin tiêu đề mã hóa dữ liệu đích. Thông tin tiêu đề mã hóa dữ liệu sai phân được sử dụng bởi thiết bị đầu cuối di động để thu được dữ liệu sai phân thông qua hoạt động giải mã, và thông tin tiêu đề mã hóa dữ liệu đích được sử dụng bởi thiết bị đầu cuối di động để xác minh chữ ký của dữ liệu được mã hóa sai phân. Dữ liệu được mã hóa sai phân là dữ liệu thu được sau khi dữ liệu sai phân được mã hóa, hoặc dữ liệu được mã hóa sai phân là dữ liệu thu được sau khi dữ liệu sai phân được mã hóa và ký. Sau đó, thiết bị đầu cuối di động giải mã dữ liệu được mã hóa sai phân dựa trên thông tin tiêu đề mã hóa dữ liệu sai phân, để thu được dữ liệu sai phân. Tiếp theo, thiết bị đầu cuối di động thu được phiên bản cũ của dữ liệu ban đầu của phần mềm hệ thống. Tiếp theo, thiết bị đầu cuối di động tích hợp phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu. Cuối cùng, thiết bị đầu cuối di động nâng cấp phiên bản cũ của phần mềm hệ thống cho phiên bản mới bằng cách sử dụng phiên bản mới của dữ liệu ban đầu.

Máy chủ thứ nhất theo phương án này của sáng chế là máy chủ OTA. Theo phương án thực hiện này của sáng chế, khi tạo ra dữ liệu OTA, máy chủ thứ nhất (cụ thể là, máy chủ OTA) có thể tạo ra dữ liệu sai phân dựa trên phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu mà thu được thông qua hoạt động giải mã, và gửi dữ liệu OTA bao gồm dữ liệu sai phân tới thiết bị đầu cuối di động. Thiết bị đầu cuối di động giải mã dữ liệu OTA để thu được dữ liệu sai phân và phiên bản cũ được lưu trữ cục bộ của dữ liệu ban đầu, và tích hợp dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu để thu được phiên bản mới của dữ liệu ban đầu. Do máy chủ OTA thực hiện phép sai phân chỉ trên phiên bản mới của dữ liệu ban đầu và phiên bản cũ của dữ liệu ban đầu, dữ liệu chữ ký của phiên bản mới của dữ liệu ban đầu được loại bỏ khỏi dữ liệu sai phân, sao cho kích thước của dữ liệu OTA được giảm, và việc nâng cấp nhanh của phần mềm hệ thống trên thiết bị đầu cuối di động được thực hiện.

Theo một phương án, trước khi thiết bị đầu cuối di động nhận dữ liệu OTA được gửi bởi máy chủ thứ nhất, phương pháp này còn bao gồm các bước: Thiết bị đầu cuối di động gửi phiên bản cũ của thông tin của phần mềm hệ thống tới máy chủ thứ nhất. Phiên bản cũ của thông tin của phần mềm hệ thống được sử dụng bởi máy chủ thứ nhất để xác định phiên bản cũ của dữ liệu được mã hóa từ một hoặc nhiều phiên bản lịch sử của dữ liệu được mã hóa. Nói theo cách khác, do máy chủ thứ nhất có thể có nhiều phiên bản lịch sử của dữ liệu của phần mềm hệ thống, thiết bị đầu cuối di động gửi phiên bản cũ của thông tin tới máy chủ thứ nhất, sao cho máy chủ thứ nhất có thể xác định phiên bản cũ của dữ liệu được mã hóa từ nhiều phiên bản lịch sử của dữ liệu, và tạo ra dữ liệu sai phân trên phiên bản cũ, để đảm bảo rằng dữ liệu sai phân được tạo ra dựa trên phiên bản cũ của phần mềm hệ thống trên thiết bị đầu cuối di động và phiên bản mới được phát hành mới bởi nhà cung cấp, và đảm bảo sự chính xác của dữ liệu sai phân dùng cho việc nâng cấp phần mềm hệ thống.

Theo một phương án, thiết bị đầu cuối di động bao gồm môi trường thực thi tin cậy (TEE: trusted execution environment) và môi trường thực thi phong phú (REE: rich execution environment). TA ứng dụng tin cậy chạy trong TEE, và CA ứng dụng khách thứ nhất và ứng dụng cập nhật chạy trong REE.

Theo một phương án, việc thiết bị đầu cuối di động giải mã dữ liệu được mã hóa sai phân dựa trên thông tin tiêu đề mã hóa dữ liệu sai phân, để thu được dữ liệu

sai phân bao gồm: Trước tiên, ứng dụng cập nhật trích xuất dữ liệu được mã hóa sai phân và thông tin tiêu đề mã hóa dữ liệu sai phân từ dữ liệu OTA. Thông tin tiêu đề mã hóa dữ liệu sai phân bao gồm thuật toán mã hóa của dữ liệu được mã hóa sai phân và giá trị ngẫu nhiên mã hóa được sử dụng trong hoạt động mã hóa. Sau đó, ứng dụng cập nhật chuyển tiếp dữ liệu được mã hóa sai phân, thuật toán mã hóa của dữ liệu được mã hóa sai phân, và giá trị ngẫu nhiên mã hóa được sử dụng trong hoạt động mã hóa tới TA bằng cách sử dụng CA thứ nhất. Cuối cùng, TA giải mã dữ liệu được mã hóa sai phân dựa trên thông tin tiêu đề mã hóa dữ liệu sai phân, để thu được dữ liệu sai phân. Theo cách này, thiết bị đầu cuối di động giải mã dữ liệu được mã hóa sai phân bằng cách sử dụng TA trong TEE, để thu được dữ liệu sai phân, để đảm bảo tính bảo mật của dữ liệu được mã hóa sai phân.

Theo một phương án, khi thông tin tiêu đề bao gồm thông tin tiêu đề mã hóa dữ liệu sai phân và thông tin tiêu đề mã hóa dữ liệu đích, trước khi thiết bị đầu cuối di động tích hợp phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu, phương pháp này còn bao gồm các bước: Trước tiên, ứng dụng cập nhật trích xuất thông tin tiêu đề mã hóa dữ liệu đích từ thông tin tiêu đề, trong đó thông tin tiêu đề mã hóa dữ liệu đích bao gồm thông tin thuật toán tóm lược thông điệp của dữ liệu được mã hóa sai phân và thông tin thuật toán mã hóa dùng cho chữ ký. Sau đó, ứng dụng cập nhật chuyển tiếp thông tin thuật toán tóm lược thông điệp của dữ liệu được mã hóa sai phân và thông tin thuật toán mã hóa dùng cho chữ ký tới TA bằng cách sử dụng CA thứ nhất. Cuối cùng, TA xác minh chữ ký của dữ liệu được mã hóa sai phân dựa trên thông tin chữ ký. Nếu việc xác minh thành công, thiết bị đầu cuối di động tích hợp phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu. Theo cách này, trước khi TA tích hợp phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu, TA thứ nhất xác minh chữ ký của dữ liệu được mã hóa sai phân, để xác định rằng dữ liệu được mã hóa sai phân không được điều chỉnh. Điều này đảm bảo tính toàn vẹn của dữ liệu sai phân được sử dụng để tích hợp phiên bản mới của dữ liệu ban đầu.

Theo một phương án, việc thiết bị đầu cuối di động thu được phiên bản cũ của dữ liệu ban đầu của phần mềm hệ thống bao gồm: Ứng dụng cập nhật chuyển tiếp phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống tới TA bằng cách sử

dụng CA thứ nhất. TA giải mã phiên bản cũ của dữ liệu được mã hóa để thu được phiên bản cũ của dữ liệu ban đầu. Theo cách này, phiên bản cũ của dữ liệu được mã hóa được giải mã bằng cách sử dụng TA, để đảm bảo tính bảo mật trong quy trình giải mã của phiên bản cũ của dữ liệu được mã hóa.

Theo một phương án, phương pháp này còn bao gồm các bước: TA gửi phiên bản cũ của dữ liệu ban đầu và dữ liệu sai phân tới CA thứ nhất. Việc thiết bị đầu cuối di động tích hợp phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu bao gồm: CA thứ nhất tích hợp phiên bản mới của dữ liệu ban đầu dựa trên phiên bản cũ của dữ liệu ban đầu và dữ liệu sai phân. Theo cách này, quy trình để tích hợp phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu được thực hiện bởi CA thứ nhất, sao cho CA thứ nhất có thể chia sẻ một phần của tác vụ xử lý dữ liệu của TA, để cải thiện tốc độ của việc nâng cấp phần mềm hệ thống trên thiết bị đầu cuối di động trong khi đảm bảo tính bảo mật dữ liệu trong quy trình nâng cấp phần mềm hệ thống.

Theo một phương án, phiên bản cũ của dữ liệu ban đầu bao gồm phiên bản cũ của khối dữ liệu tệp, phiên bản cũ của khối dữ liệu hàm băm, và phiên bản cũ của khối dữ liệu kiểm tra. Phiên bản cũ của khối dữ liệu kiểm tra là dữ liệu chữ ký của giá trị hàm băm gốc trong phiên bản cũ của khối dữ liệu hàm băm, và phiên bản cũ của khối dữ liệu hàm băm được sử dụng để kiểm tra phiên bản cũ của khối dữ liệu tệp;

Việc TA tích hợp phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu bao gồm: Trước tiên, TA tích hợp phiên bản mới của khối dữ liệu tệp và phiên bản mới của khối dữ liệu kiểm tra dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu; và sau đó, TA tính toán cây hàm băm của phiên bản mới của khối dữ liệu tệp, và tạo ra phiên bản mới của khối dữ liệu hàm băm, để thu được phiên bản mới của dữ liệu ban đầu. Phiên bản mới của dữ liệu ban đầu bao gồm phiên bản mới của khối dữ liệu tệp, phiên bản mới của khối dữ liệu hàm băm, và phiên bản mới của khối dữ liệu kiểm tra. Phiên bản mới của khối dữ liệu kiểm tra là dữ liệu chữ ký của giá trị hàm băm gốc trong phiên bản mới của khối dữ liệu hàm băm, và phiên bản mới của khối dữ liệu hàm băm được sử dụng để kiểm tra phiên bản mới của khối dữ liệu tệp. Theo cách này, khi tạo ra dữ liệu sai phân, máy chủ thứ nhất (máy chủ OTA) loại bỏ các khối dữ liệu tương ứng với cây hàm băm, và duy trì phiên

bản mới của khối dữ liệu kiểm tra và phiên bản mới của khối dữ liệu tệp, và sau đó khi thiết bị đầu cuối di động hồi phục dữ liệu sai phân, máy chủ thứ nhất (máy chủ OTA) tạo ra, bằng cách sử dụng giá trị ngẫu nhiên kiểm tra trong phiên bản mới được hồi phục của khối dữ liệu tệp và phiên bản mới của khối dữ liệu kiểm tra, khối dữ liệu hàm băm tương ứng với phiên bản mới của khối dữ liệu tệp, và lưu trữ khối dữ liệu hàm băm vào trong phương tiện lưu trữ. Phiên bản mới của khối dữ liệu hàm băm được loại bỏ khỏi dữ liệu OTA, và phiên bản mới của khối dữ liệu hàm băm được tạo ra trên thiết bị đầu cuối di động được sử dụng để làm giảm lượng dữ liệu của dữ liệu cập nhật phần mềm hệ thống (dữ liệu OTA) được phát hành bởi máy chủ thứ nhất.

Theo khía cạnh thứ ba, sáng chế đề xuất máy chủ, bao gồm bộ nhớ, bộ thu phát, và ít nhất một bộ xử lý. Bộ nhớ lưu trữ mã chương trình. Bộ nhớ, bộ thu phát, và ít nhất một bộ xử lý truyền thông với nhau. Bộ xử lý chạy mã để ra lệnh cho máy chủ thực hiện phương pháp gửi dữ liệu bằng công nghệ qua vô tuyến OTA theo khía cạnh thứ nhất hoặc phương án thực hiện có thể có bất kỳ của khía cạnh thứ nhất.

Theo khía cạnh thứ tư, sáng chế đề xuất thiết bị đầu cuối di động, bao gồm bộ nhớ, bộ thu phát, và ít nhất một bộ xử lý. Bộ nhớ lưu trữ mã chương trình. Bộ nhớ, bộ thu phát, và ít nhất một bộ xử lý truyền thông với nhau. Bộ xử lý chạy mã để ra lệnh cho thiết bị đầu cuối di động thực hiện phương pháp nâng cấp qua vô tuyến theo khía cạnh thứ hai hoặc phương án thực hiện có thể có bất kỳ của khía cạnh thứ hai.

Theo khía cạnh thứ năm, phương án thực hiện của sáng chế đề xuất phương tiện lưu trữ đọc được bằng máy tính, bao gồm các lệnh máy tính. Khi các lệnh máy tính được chạy trên máy chủ, máy chủ thực hiện phương pháp gửi dữ liệu bằng công nghệ qua vô tuyến OTA theo khía cạnh thứ nhất hoặc phương án thực hiện có thể có bất kỳ của khía cạnh thứ nhất.

Theo khía cạnh thứ sáu, phương án thực hiện của sáng chế đề xuất phương tiện lưu trữ đọc được bằng máy tính, bao gồm các lệnh máy tính. Khi các lệnh máy tính được chạy trên thiết bị đầu cuối di động, thiết bị đầu cuối di động thực hiện phương pháp gửi dữ liệu bằng công nghệ qua vô tuyến OTA theo khía cạnh thứ hai hoặc phương án thực hiện có thể có bất kỳ của khía cạnh thứ hai.

Mô tả vắn tắt các hình vẽ

FIG.1 là hình vẽ giản lược của kiến trúc hệ thống theo một phương án thực hiện của sáng chế;

FIG.2 là hình vẽ kết cấu giản lược của thiết bị đầu cuối di động theo một phương án thực hiện của sáng chế;

FIG.3 là hình vẽ kết cấu giản lược của máy chủ OTA theo một phương án thực hiện của sáng chế;

FIG.4a là lưu đồ giản lược của thuật toán sai phân theo một phương án thực hiện của sáng chế;

FIG.4b là lưu đồ giản lược của thuật toán sai phân khác theo một phương án thực hiện của sáng chế;

FIG.5a là lưu đồ giản lược của phương pháp nâng cấp qua vô tuyến theo một phương án thực hiện của sáng chế;

FIG.5b là lưu đồ giản lược của phương pháp nâng cấp qua vô tuyến theo phương án khác của sáng chế;

FIG.6A và FIG.6B là lưu đồ giản lược của phương pháp nâng cấp qua vô tuyến theo phương án khác của sáng chế;

FIG.7a và FIG.7b là lưu đồ giản lược của việc trích xuất dữ liệu theo một phương án thực hiện của sáng chế;

FIG.8 là lưu đồ giản lược của việc tạo ra dữ liệu sai phân theo một phương án thực hiện của sáng chế;

FIG.9 là lưu đồ giản lược của việc tạo ra dữ liệu OTA theo một phương án thực hiện của sáng chế;

FIG.10A và FIG.10B là lưu đồ giản lược của việc nâng cấp phần mềm hệ thống bằng thiết bị đầu cuối di động theo một phương án thực hiện của sáng chế;

FIG.11A và FIG.11B là lưu đồ giản lược của việc nâng cấp phần mềm hệ thống bằng thiết bị đầu cuối di động theo phương án khác của sáng chế;

FIG.12 là lưu đồ giản lược của việc tạo ra khối dữ liệu hàm băm theo một phương án thực hiện của sáng chế; và

FIG.13A và FIG.13B là lưu đồ giản lược của phương pháp nâng cấp qua vô

tuyến theo phương án khác của sáng chế.

Mô tả chi tiết sáng chế

Các giải pháp kỹ thuật theo các phương án thực hiện của sáng chế được mô tả rõ ràng một cách chi tiết dưới đây tham chiếu đến các hình vẽ kèm theo. Trong các mô tả các phương án thực hiện của sáng chế, dấu "/" có nghĩa là "hoặc" trừ khi có mô tả cụ thể khác. Ví dụ, A/B có thể là A hoặc B. Trong bản mô tả này, "và/hoặc" chỉ mô tả mối quan hệ kết hợp để mô tả các đối tượng liên quan và thể hiện rằng các mối quan hệ này có thể tồn tại. Ví dụ, A và/hoặc B có thể thể hiện ba trường hợp sau đây: Chỉ có A tồn tại, cả A và B đều tồn tại, và chỉ có B tồn tại. Ngoài ra, trong các mô tả theo các phương án thực hiện của sáng chế, "nhiều" có nghĩa là hai hoặc nhiều hơn hai.

Các từ "thứ nhất" và "thứ hai" dưới đây chỉ được dùng nhằm mục đích mô tả, và sẽ không được hiểu là sự chỉ định hoặc sự ngầm ý về mức độ quan trọng tương đối hoặc sự chỉ định ngầm ý về số lượng các dấu hiệu kỹ thuật được chỉ định. Vì vậy, dấu hiệu được giới hạn bởi từ "thứ nhất" hoặc "thứ hai" có thể bao gồm một cách tường minh hoặc không tường minh một hoặc nhiều dấu hiệu. Trong các mô tả của các phương án thực hiện của sáng chế, trừ khi có mô tả cụ thể khác, "nhiều" có nghĩa là hai hoặc nhiều hơn hai.

Tham chiếu đến FIG.1, FIG.1 là hình vẽ thể hiện kiến trúc mạng theo phương án thực hiện của sáng chế. Như được thể hiện trên FIG.1, kiến trúc mạng 10 có thể bao gồm máy chủ công nghệ qua vô tuyến (over-the-air technology, OTA) 300, thiết bị đầu cuối di động 200, và liên kết truyền thông không dây. Ví dụ, thiết bị đầu cuối di động 200 có thể truyền thông với máy chủ OTA 300 bằng cách sử dụng mạng truyền thông di động (ví dụ, mạng 2G\3G\4G\5G), hoặc có thể truyền thông với máy chủ OTA 300 bằng cách sử dụng mạng cục bộ không dây (Wireless Local Area Network, WLAN).

Thiết bị đầu cuối di động 200 có thể bao gồm nhưng không bị giới hạn ở máy tính cá nhân, điện thoại thông minh, TV thông minh, máy tính bảng, thiết bị hỗ trợ kỹ thuật số cá nhân, hoặc tương tự.

Tham chiếu đến FIG.2, FIG.2 là hình vẽ kết cấu giản lược của môi trường thực

thi tin cậy của thiết bị đầu cuối di động theo một phương án thực hiện của sáng chế. Thiết bị đầu cuối di động 200 có thể bao gồm hai môi trường ứng dụng: môi trường thực thi phong phú (rich execution environment, REE) và môi trường thực thi tin cậy (trusted execution environment, TEE). Ứng dụng chạy trong REE có thể được gọi là ứng dụng khách (client application, CA), và ứng dụng khách trong REE có thể bao gồm ứng dụng cập nhật và CA thứ nhất. Ứng dụng chạy trong TEE có thể được gọi là ứng dụng tin cậy (TEE application: ứng dụng TEE, TA). Ứng dụng cập nhật có thể được tạo cấu hình để: tải lên số phiên bản của phần mềm hệ thống trên thiết bị đầu cuối di động cho máy chủ OTA, và tải xuống, từ máy chủ OTA, dữ liệu OTA dùng cho việc nâng cấp phần mềm hệ thống. CA thứ nhất có thể được sử dụng để cung cấp chức năng chuyển tiếp dữ liệu cho ứng dụng cập nhật và TA, để thực hiện sự truyền thông gián tiếp của ứng dụng cập nhật trong TA. TA có thể được sử dụng cho việc mã hóa, giải mã, ký, hồi phục sai phân và hoạt động tương tự cho dữ liệu. Hệ điều hành chạy trong REE có thể được gọi là hệ điều hành môi trường thực thi phong phú (rich execution environment operating system, REE OS), và hệ điều hành chạy trong TEE có thể được gọi là hệ điều hành môi trường thực thi tin cậy (trusted execution environment operating system, TEE OS). TEE là môi trường vận hành bảo mật chạy trong bộ xử lý chính. Quá trình khởi động bảo mật của TEE cần được xác minh, và quá trình khởi động bảo mật của TEE được tách biệt với REE. Các chương trình ứng dụng chạy trong TEE là độc lập với nhau, và các chương trình ứng dụng không thể truy nhập lẫn nhau mà không có ủy quyền, để đảm bảo rằng việc xử lý của các tài nguyên và dữ liệu của các chương trình ứng dụng trong TEE được thực hiện trong môi trường tin cậy, nhờ đó cung cấp dịch vụ bảo mật cho hệ điều hành REE. TEE có không gian thực thi của chính nó, và có mức bảo mật cao hơn hệ điều hành REE. Ngoài ra, TEE không là chip bảo mật vật lý độc lập, nhưng có kiến trúc bảo mật mà chồng lên kiến trúc phần cứng của bộ xử lý ứng dụng được sử dụng hiện nay. Các tài nguyên phần mềm và phần cứng mà có thể được truy nhập bởi TEE được tách biệt khỏi hệ điều hành REE để cung cấp sự cách ly phần cứng hỗ trợ. Giao diện máy khách môi trường thực thi tin cậy (TEE Client API) và giao diện nội bộ môi trường thực thi tin cậy (TEE Internal API) có thể sử dụng một cách tương ứng tiêu chuẩn TEE Client API V1.0 và tiêu chuẩn TEE Internal API V1.0.

CA thứ nhất và TA chia sẻ bộ nhớ. Nếu CA thứ nhất cần truyền thông với TA, CA thứ nhất có thể áp dụng cho TEE OS để thiết lập phiên (session) với TA được yêu cầu. Sau khi việc thiết lập phiên giữa CA thứ nhất và TA được hoàn thành, CA thứ nhất có thể gửi yêu cầu xử lý và dữ liệu mà cần được xử lý tới TA bằng cách sử dụng bộ nhớ được chia sẻ. Sau khi thu được yêu cầu xử lý và dữ liệu mà cần được xử lý của CA thứ nhất từ bộ nhớ được chia sẻ, TA có thể thực thi yêu cầu xử lý trong môi trường TEE, và lưu trữ kết quả xử lý đã thu được vào trong bộ nhớ được chia sẻ. CA thứ nhất có thể thu được kết quả xử lý của TA từ bộ nhớ được chia sẻ. Sau khi CA thứ nhất thu được kết quả xử lý, nếu TA không cần tiếp tục xử lý, CA thứ nhất có thể khởi đầu yêu cầu đóng phiên (close session) tới TEE OS. Sau khi nhận yêu cầu đóng phiên (close session), TEE OS có thể phục hồi các tài nguyên liên quan của TA.

Phần cứng nền tảng (platform hardware) của thiết bị đầu cuối di động 200 có thể bao gồm bộ xử lý 201, nguồn điện 202, giao diện truyền thông 203, màn hình chạm 204, phương tiện lưu trữ 208, và mạch âm thanh 209. Phương tiện lưu trữ 208 được tạo cấu hình để lưu trữ hệ điều hành 205, dữ liệu 206, và chương trình ứng dụng 207 của thiết bị đầu cuối di động 200. Ví dụ, vùng được bảo vệ trong phương tiện lưu trữ 208 có thể lưu trữ hệ điều hành môi trường thực thi tin cậy (trusted execution environment operating system, TEE OS) và ứng dụng (TA) trong TEE. Vùng không được bảo vệ trong phương tiện lưu trữ 208 có thể lưu trữ hệ điều hành môi trường thực thi phong phú (rich execution environment operating system, REE OS) và ứng dụng (CA) trong REE. Chương trình ứng dụng 207 bao gồm ứng dụng khách (CA) trong REE và ứng dụng tin cậy (TA) trong TEE. Hệ điều hành 205 bao gồm REE OS và TEE OS. REE OS có thể bao gồm hệ điều hành như hệ điều hành Android và hệ điều hành EMUI. Dữ liệu 206 có thể bao gồm dữ liệu OTA, dữ liệu được mã hóa phiên bản của phần mềm hệ thống, và tương tự theo phương án này của sáng chế. Bộ xử lý 201 có thể là cụm xử lý trung tâm (central processing unit, CPU), bộ xử lý đa năng, bộ xử lý tín hiệu số (digital signal processing, DSP), mạch tích hợp chuyên dụng (application-specific integrated circuit, ASIC), mảng cổng khả lập trình dạng trường (field programmable gate array, FPGA), hoặc thiết bị logic khả lập trình khác, thiết bị logic tranzito, thành phần phần cứng, hoặc sự kết hợp bất kỳ của chúng.

Phương tiện lưu trữ 208 có thể bao gồm bộ nhớ truy cập ngẫu nhiên tốc độ cao,

và có thể còn bao gồm bộ nhớ bất khả biến, ví dụ, ít nhất một thiết bị nhớ tác động nhanh (flash memory), hoặc thiết bị lưu trữ thể rắn khả biến khác. Các chương trình và các luồng dữ liệu đã nhận của phương pháp nâng cấp qua vô tuyến mà được đề xuất theo phương án thực hiện này của sáng chế được lưu trữ trong bộ nhớ, và có thể được goi từ phương tiện lưu trữ 208 bởi bộ xử lý 201 khi được yêu cầu để được sử dụng.

Thiết bị đầu cuối di động 200 có thể còn bao gồm giao diện truyền thông 203, màn hình chạm 204, và mạch âm thanh 209. Giao diện truyền thông 203 có thể được sử dụng bởi thiết bị đầu cuối di động 200 để truyền thông với máy chủ 300 hoặc thiết bị đầu cuối khác. Mạch âm thanh 209 có thể được tạo cấu hình để chuyển đổi thông tin âm thanh dạng số thành tín hiệu âm thanh dạng tương tự cho đầu ra, hoặc có thể được tạo cấu hình để chuyển đổi đầu vào âm thanh dạng tương tự thành tín hiệu âm thanh dạng số. Mạch âm thanh 209 có thể còn được tạo cấu hình để mã hóa và giải mã tín hiệu âm thanh. Theo một số phương án thực hiện, mạch âm thanh 209 có thể được bố trí trong bộ xử lý 201, hoặc một số môđun chức năng của mạch âm thanh 209 có thể được bố trí trong bộ xử lý 201. Màn hình chạm 204 có thể bao gồm bộ cảm biến chạm và màn hiển thị. Bộ cảm biến chạm có thể truyền hoạt động chạm được phát hiện tới bộ xử lý 201, để xác định loại sự kiện chạm. Đầu ra nhìn thấy liên quan đến hoạt động chạm có thể được cung cấp thông qua màn hiển thị. Theo một số phương án thực hiện, bộ cảm biến chạm có thể được bố trí theo cách thay thế trên bề mặt của thiết bị đầu cuối di động 200, và nằm ở vị trí khác với vị trí của màn hiển thị. Người có hiểu biết trung bình trong lĩnh vực tương ứng có thể hiểu rằng kết cấu máy chủ được thể hiện trên FIG.2 không nên hiểu là sự giới hạn. Theo phương án thực hiện cụ thể, máy chủ có thể bao gồm nhiều hoặc ít hơn các thành phần so với thiết bị đầu cuối di động 200 được thể hiện trên FIG.2, hoặc kết hợp một số thành phần, hoặc có các cách bố trí thành phần khác nhau.

Tham chiếu đến FIG.3, FIG.3 là hình vẽ kết cấu giản lược của máy chủ theo một phương án thực hiện của sáng chế. Như được thể hiện trên FIG.3, máy chủ 300 có thể bao gồm một hoặc nhiều bộ xử lý 301 (ví dụ, bộ xử lý trung tâm (central processing unit, CPU)) và phương tiện lưu trữ 308. Phương tiện lưu trữ 308 có thể lưu trữ một hoặc nhiều chương trình ứng dụng 307, dữ liệu 306, và hệ điều hành 305. Dữ liệu 306 có thể bao gồm dữ liệu phiên bản của phần mềm hệ thống tham gia vào

phương pháp nâng cấp qua vô tuyến mà được đề xuất theo phương án thực hiện này của sáng chế, ví dụ, dữ liệu như phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa của phần mềm hệ thống, và dữ liệu OTA được tạo ra dựa trên phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa. Phương tiện lưu trữ 308 có thể là bộ lưu trữ tạm thời hoặc bộ lưu trữ không đổi. Bộ xử lý 301 có thể được tạo cấu hình để truyền thông với phương tiện lưu trữ 308.

Phương tiện lưu trữ 308 có thể được tạo cấu hình để lưu trữ chương trình phần mềm và môđun. Bộ xử lý 301 thực thi các ứng dụng chức năng khác nhau và xử lý dữ liệu của máy chủ OTA 300 bằng cách chạy chương trình phần mềm và môđun chứa trong phương tiện lưu trữ 308. Phương tiện lưu trữ 308 có thể chủ yếu bao gồm vùng lưu trữ chương trình và vùng lưu trữ dữ liệu. Vùng lưu trữ chương trình có thể lưu trữ hệ điều hành, chương trình ứng dụng được yêu cầu bởi ít nhất một chức năng, và tương tự. Vùng lưu trữ dữ liệu có thể lưu trữ dữ liệu được tạo ra dựa trên sử dụng máy chủ OTA (ví dụ, lệnh tải lên dữ liệu), và tương tự. Ngoài ra, phương tiện lưu trữ 308 có thể bao gồm bộ nhớ truy cập ngẫu nhiên tốc độ cao, và có thể còn bao gồm bộ nhớ bất khả biến, ví dụ, ít nhất một thiết bị lưu trữ đĩa từ, thiết bị nhớ tác động nhanh, hoặc thiết bị lưu trữ thể rắn khả biến khác. Các chương trình và các luồng dữ liệu đã nhận theo phương pháp nâng cấp bằng công nghệ qua vô tuyến mà được đề xuất theo phương án thực hiện này của sáng chế được lưu trữ trong bộ nhớ, và được goi từ phương tiện lưu trữ 308 bởi bộ xử lý 301 khi được yêu cầu để được sử dụng.

Máy chủ OTA 300 có thể còn bao gồm một hoặc nhiều nguồn điện 302, một hoặc nhiều giao diện mạng có dây hoặc không dây 303, một hoặc nhiều giao diện vào/ra 304, và/hoặc một hoặc nhiều hệ điều hành 305, ví dụ, Windows Serve, Mac OS X, Unix, Linux, và FreeBSD. Người có hiểu biết trung bình trong lĩnh vực tương ứng có thể hiểu rằng kết cấu máy chủ được thể hiện trên FIG.3 không nên hiểu là sự giới hạn. Theo phương án thực hiện cụ thể, máy chủ có thể bao gồm nhiều hoặc ít hơn các thành phần so với máy chủ được thể hiện trên FIG.3, hoặc kết hợp một số thành phần, hoặc có các cách bố trí thành phần khác nhau.

Thiết bị đầu cuối di động được mô tả theo phương án này của sáng chế có thể là thiết bị đầu cuối di động mà các môi trường bảo mật của cả REE và TEE, như điện thoại thông minh, TV thông minh, máy tính bảng, hoặc thiết bị hỗ trợ kỹ thuật số cá

nhân

Trong quy trình nâng cấp OTA trong lĩnh vực liên quan, do nhà cung cấp của phần mềm hệ thống cần đảm bảo tính bảo mật của dữ liệu phiên bản của phần mềm hệ thống trong quy trình nâng cấp phần mềm hệ thống, bộ gửi của dữ liệu phiên bản cần mã hóa và ký dữ liệu phiên bản trước khi gửi dữ liệu phiên bản tới bộ nhận của dữ liệu phiên bản. Do đó, mỗi khi thiết bị đầu cuối của nhà cung cấp tạo ra một phiên bản của phần mềm hệ thống, thiết bị đầu cuối cần mã hóa và ký dữ liệu phiên bản của phần mềm hệ thống, để đảm bảo tính bảo mật của dữ liệu phiên bản của phần mềm hệ thống. Dữ liệu phiên bản của phần mềm hệ thống thu được từ thiết bị đầu cuối của nhà cung cấp bởi máy chủ OTA được mã hóa và ký. Máy chủ OTA lưu trữ các phiên bản khác nhau của dữ liệu được mã hóa của phần mềm hệ thống (bao gồm phiên bản mới của dữ liệu được mã hóa và nhiều phiên bản lịch sử của dữ liệu được mã hóa). Máy chủ OTA xác định, dựa trên số phiên bản hiện tại của phần mềm hệ thống trên thiết bị đầu cuối di động, phiên bản cũ của dữ liệu được mã hóa mà phù hợp với số phiên bản hiện tại từ nhiều phiên bản lịch sử của dữ liệu được mã hóa được lưu trữ trên máy chủ OTA. Máy chủ OTA thực hiện phép sai phân một cách trực tiếp trên phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa để thu được kết quả sai phân. Phép sai phân có thể được thực hiện một cách trực tiếp trên phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa của nhóm tệp không nén bằng cách sử dụng thuật toán sai phân nhị phân (binary difference, bsdiff), và phép sai phân có thể được thực hiện phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa của nhóm tệp nén bằng cách sử dụng thuật toán imgdiff. Kết quả sai phân được gửi tới thiết bị đầu cuối di động. Sau khi nhận kết quả sai phân, thiết bị đầu cuối di động hồi phục phiên bản mới của dữ liệu được mã hóa của phần mềm hệ thống bằng cách sử dụng kết quả sai phân và phiên bản cũ của dữ liệu được mã hóa tương ứng với số phiên bản hiện tại của phần mềm hệ thống, và lưu trữ phiên bản mới của dữ liệu được mã hóa vào trong phương tiện lưu trữ của thiết bị đầu cuối di động.

Như được thể hiện trên FIG.4a, khi thực hiện phép sai phân trên phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa của nhóm tệp không nén bằng cách sử dụng thuật toán bsdiff, máy chủ OTA có thể so sánh một cách trực tiếp phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã

hóa để xuất ra kết quả sai phân. Máy chủ OTA gửi kết quả sai phân tới thiết bị đầu cuối di động. Sau khi nhận kết quả sai phân, thiết bị đầu cuối di động có thể hồi phục phiên bản mới của dữ liệu được mã hóa bằng cách sử dụng thuật toán bspatch dựa trên phiên bản cũ của dữ liệu được mã hóa được lưu trữ trên thiết bị đầu cuối di động và kết quả sai phân đã nhận được, và ghi phiên bản mới được hồi phục của dữ liệu được mã hóa vào trong phương tiện lưu trữ.

Như được thể hiện trên FIG.4b, khi thực hiện phép sai phân trên phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa của nhóm tệp nén (định dạng nén có thể là ".Apk", ".zip", hoặc ".jar") bằng cách sử dụng thuật toán imgdiff, trước tiên máy chủ OTA có thể giải nén phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa của nhóm tệp nén thành các tệp tin không nén, và sau đó tạo ra kết quả sai phân bằng cách sử dụng thuật toán bsdiff. Máy chủ OTA gửi kết quả sai phân tới thiết bị đầu cuối di động. Sau khi nhận kết quả sai phân, trước tiên thiết bị đầu cuối di động có thể đọc phiên bản cũ của dữ liệu được mã hóa của nhóm tệp nén được lưu trữ trên thiết bị đầu cuối di động, và giải nén phiên bản cũ của dữ liệu được mã hóa thành các tệp tin không nén. Sau đó, máy chủ OTA hồi phục phiên bản mới của dữ liệu được mã hóa bằng cách sử dụng thuật toán bspath dựa trên phiên bản cũ của dữ liệu được mã hóa và kết quả sai phân, và ghi phiên bản mới được hồi phục của dữ liệu được mã hóa vào trong phương tiện lưu trữ.

Do phiên bản cũ của dữ liệu được mã hóa bao gồm dữ liệu ciphertext thu được sau khi phiên bản cũ của dữ liệu ban đầu của phần mềm hệ thống được mã hóa và dữ liệu chữ ký trong phiên bản cũ của dữ liệu ban đầu (bao gồm chuỗi ký tự thu được sau khi phân loại phiên bản cũ của dữ liệu ban đầu được mã hóa), phiên bản mới của dữ liệu được mã hóa bao gồm dữ liệu ciphertext thu được sau khi phiên bản mới của dữ liệu ban đầu của phần mềm hệ thống được mã hóa và dữ liệu chữ ký trong phiên bản mới của dữ liệu ban đầu (bao gồm chuỗi ký tự thu được sau khi bản tóm lược của phiên bản mới của dữ liệu ban đầu được mã hóa). Phiên bản cũ lớn hơn của dữ liệu ban đầu chỉ báo bản tóm lược lớn hơn của phiên bản cũ của dữ liệu ban đầu. Tương tự, phiên bản mới lớn hơn của dữ liệu ban đầu chỉ báo bản tóm lược lớn hơn của phiên bản mới của dữ liệu ban đầu. Do đó, khi thực hiện phép sai phân một cách trực tiếp trên phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa,

máy chủ OTA cũng thực hiện phép sai phân trên bản tóm lược của phiên bản cũ của dữ liệu ban đầu và bản tóm lược của phiên bản mới của dữ liệu ban đầu. Do bản tóm lược của phiên bản cũ của dữ liệu ban đầu khác với bản tóm lược của phiên bản mới của dữ liệu ban đầu, dữ liệu sai phân bao gồm một số dữ liệu mà nằm trong phiên bản mới của dữ liệu ban đầu và khác với phiên bản cũ của dữ liệu ban đầu, và dữ liệu chữ ký của phiên bản mới của dữ liệu ban đầu.

Tuy nhiên, thiết bị đầu cuối di động có thể hồi phục phiên bản mới của dữ liệu ban đầu chỉ bằng cách sử dụng một số dữ liệu khác nhau trong dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu của phần mềm hệ thống được lưu trữ trên thiết bị đầu cuối di động, và khi truyền dữ liệu sai phân tới thiết bị đầu cuối di động, máy chủ OTA không chỉ truyền một số dữ liệu khác nhau, mà còn truyền dữ liệu chữ ký của phiên bản mới của dữ liệu ban đầu. Dữ liệu chữ ký của phiên bản mới của dữ liệu ban đầu cũng cần để chiếm một lượng dữ liệu của một số gói cập nhật. Phiên bản mới lớn hơn của dữ liệu ban đầu chỉ báo dữ liệu chữ ký lớn hơn của phiên bản mới của dữ liệu ban đầu. Do đó, ngay cả khi một số dữ liệu khác nhau giữa phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu là tương đối nhỏ, dữ liệu được gửi bởi máy chủ OTA tới thiết bị đầu cuối di động cho việc nâng cấp phần mềm hệ thống cũng bao gồm dữ liệu chữ ký tương đối lớn của phiên bản mới của dữ liệu ban đầu, thời gian để tải xuống dữ liệu dùng cho việc nâng cấp phần mềm hệ thống bằng thiết bị đầu cuối di động bị kéo dài, và trải nghiệm người dùng bị ảnh hưởng nghiêm trọng.

Vì lý do này, tham chiếu đến FIG.5a, FIG.5a là lưu đồ giản lược của phương pháp nâng cấp qua vô tuyến theo một phương án thực hiện của sáng chế. Như được thể hiện trên FIG.5a, phiên bản cũ của dữ liệu được mã hóa bao gồm dữ liệu ciphertext thu được sau khi phiên bản cũ của dữ liệu ban đầu của phần mềm hệ thống được mã hóa và dữ liệu chữ ký trong phiên bản cũ của dữ liệu ban đầu (bao gồm chuỗi ký tự thu được sau khi phân loại phiên bản cũ của dữ liệu ban đầu được mã hóa), và phiên bản mới của dữ liệu được mã hóa bao gồm dữ liệu ciphertext thu được sau khi phiên bản mới của dữ liệu ban đầu của phần mềm hệ thống được mã hóa và dữ liệu chữ ký trong phiên bản mới của dữ liệu ban đầu (bao gồm chuỗi ký tự sau khi bản tóm lược của phiên bản mới của dữ liệu ban đầu được mã hóa). Máy chủ OTA có thể loại bỏ dữ liệu chữ ký trong phiên bản cũ của dữ liệu được mã hóa và dữ liệu chữ ký trong phiên

bản mới của dữ liệu được mã hóa, và thu được khóa từ máy chủ lưu trữ khóa để thu được phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu thông qua hoạt động giải mã. Sau đó, máy chủ OTA có thể thực hiện phép sai phân trên phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu, để tạo ra dữ liệu sai phân. Tiếp theo, máy chủ OTA mã hóa dữ liệu sai phân bằng cách sử dụng khóa trên máy chủ lưu trữ khóa và giá trị ngẫu nhiên mã hóa, và thực hiện việc bảo vệ chữ ký trên kết quả được mã hóa để thu được dữ liệu OTA. Máy chủ OTA có thể gửi dữ liệu OTA tới thiết bị đầu cuối di động.

Sau khi thiết bị đầu cuối di động tải xuống dữ liệu OTA, sau khi ứng dụng cập nhật trên thiết bị đầu cuối di động nhận dạng rằng dữ liệu OTA là dữ liệu được mã hóa, CA thứ nhất có thể được bắt đầu, và dữ liệu được mã hóa sai phân trong dữ liệu OTA, phiên bản cũ của dữ liệu được mã hóa được lưu trữ cục bộ trong thiết bị đầu cuối di động, và thông tin chữ ký và thông tin mã hóa của dữ liệu sai phân được chuyển tiếp tới TA thứ nhất bằng cách sử dụng CA thứ nhất. TA thứ nhất có thể giải mã phiên bản cũ của dữ liệu được mã hóa và dữ liệu được mã hóa sai phân, và hồi phục phiên bản mới của dữ liệu ban đầu. TA thứ nhất mã hóa và ký phiên bản mới của dữ liệu ban đầu, để tạo ra phiên bản mới của dữ liệu được mã hóa, và chuyển tiếp phiên bản mới của dữ liệu được mã hóa tới ứng dụng cập nhật bằng cách sử dụng CA thứ nhất. Ứng dụng cập nhật ghi phiên bản mới của dữ liệu được mã hóa vào trong phương tiện lưu trữ.

Theo một phương án, FIG.5b là lưu đồ khác của phương pháp nâng cấp qua vô tuyến theo một phương án thực hiện của sáng chế. Như được thể hiện trên FIG.5b, phương pháp được thực thi bởi máy chủ lưu trữ khóa, thiết bị đầu cuối, máy chủ OTA, và thiết bị đầu cuối di động. Thiết bị đầu cuối có thể loại bỏ dữ liệu chữ ký trong phiên bản cũ của dữ liệu được mã hóa và dữ liệu chữ ký trong phiên bản mới của dữ liệu được mã hóa và thu được khóa từ máy chủ lưu trữ khóa để thu được phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu thông qua hoạt động giải mã. Sau đó, thiết bị đầu cuối có thể thực hiện phép sai phân trên phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu, để tạo ra dữ liệu sai phân. Tiếp theo, thiết bị đầu cuối có thể mã hóa dữ liệu sai phân bằng cách sử dụng khóa trên máy chủ lưu trữ khóa và giá trị ngẫu nhiên mã hóa, và thực hiện việc bảo vệ chữ ký trên kết quả được mã hóa, để thu được dữ liệu OTA. Thiết bị đầu cuối có thể tải lên dữ liệu OTA

cho máy chủ OTA. Máy chủ OTA có thể gửi dữ liệu OTA tới thiết bị đầu cuối di động. Sau khi thiết bị đầu cuối di động tải xuống dữ liệu OTA, sau khi ứng dụng cập nhật trên thiết bị đầu cuối di động nhận dạng rằng dữ liệu OTA là dữ liệu được mã hóa, CA thứ nhất có thể được bắt đầu, và dữ liệu được mã hóa sai phân trong dữ liệu OTA, phiên bản cũ của dữ liệu được mã hóa được lưu trữ cục bộ trong thiết bị đầu cuối di động, và thông tin chữ ký và thông tin mã hóa của dữ liệu sai phân được chuyển tiếp tới TA thứ nhất bằng cách sử dụng CA thứ nhất. TA thứ nhất có thể giải mã phiên bản cũ của dữ liệu được mã hóa và dữ liệu được mã hóa sai phân, và hồi phục phiên bản mới của dữ liệu ban đầu. TA thứ nhất mã hóa và ký phiên bản mới của dữ liệu ban đầu, để tạo ra phiên bản mới của dữ liệu được mã hóa, và chuyển tiếp phiên bản mới của dữ liệu được mã hóa tới ứng dụng cập nhật bằng cách sử dụng CA thứ nhất. Ứng dụng cập nhật ghi phiên bản mới của dữ liệu được mã hóa vào phương tiện lưu trữ.

Theo phương án thực hiện này của sáng chế, trong khi đảm bảo tính bảo mật của dữ liệu trong quy trình nâng cấp phần mềm hệ thống, máy chủ OTA có thể thực hiện phép sai phân chỉ trên phiên bản mới của dữ liệu ban đầu và phiên bản cũ của dữ liệu ban đầu, và loại bỏ dữ liệu chữ ký của phiên bản mới của dữ liệu ban đầu khỏi dữ liệu sai phân, sao cho kích thước của dữ liệu được truyền bởi máy chủ OTA tới thiết bị đầu cuối di động cho việc nâng cấp phần mềm hệ thống được giảm, và việc nâng cấp nhanh của phần mềm hệ thống được thực hiện.

Phần dưới đây mô tả cụ thể phương pháp nâng cấp qua vô tuyến được thể hiện trên FIG.5a.

Theo phương án này của sáng chế, phần mềm hệ thống có thể là phần mềm hệ thống Android, phần mềm hệ điều hành cảm xúc (ví dụ, EMUI hoặc MIUI) được phát triển dựa trên hệ điều hành Android, hoặc tương tự. Theo cách thay thế, phần mềm hệ thống có thể phần mềm hệ thống như Windows Phone, IOS, hoặc tương tự. Điều này không bị giới hạn ở đây.

Máy chủ OTA có thể nhận nhiều phiên bản của dữ liệu được mã hóa của phần mềm hệ thống được gửi bởi thiết bị đầu cuối của nhà cung cấp của phần mềm hệ thống, trong đó các ngày phát hành của nhiều phiên bản của phần mềm hệ thống là khác nhau. Máy chủ OTA có thể nhận thông tin phiên bản hiện tại mà thuộc về phần mềm hệ thống trên thiết bị đầu cuối di động và được gửi bởi thiết bị đầu cuối di động. Sau khi

nhận thông tin phiên bản hiện tại, máy chủ OTA xác định xem phiên bản hiện tại của phần mềm hệ thống được sử dụng trên thiết bị đầu cuối di động có là phiên bản có ngày phát hành cuối cùng hay không, và nếu phiên bản hiện tại không là phiên bản cuối cùng, phần mềm hệ thống của thiết bị đầu cuối di động cần được nâng cấp. Máy chủ OTA có thể xác định, từ nhiều phiên bản của dữ liệu được mã hóa của phần mềm hệ thống, phiên bản cũ của dữ liệu được mã hóa tương ứng với thông tin phiên bản hiện tại và phiên bản mới của dữ liệu được mã hóa với ngày phát hành cuối cùng. Sử dụng phần mềm hệ điều hành EMUI như một ví dụ, máy chủ OTA có thể lưu trữ phiên bản 8.0 của hệ điều hành EMUI, phiên bản 8.1 của hệ điều hành EMUI, và phiên bản 8.2 của hệ điều hành EMUI mà được gửi bởi thiết bị đầu cuối của nhà cung cấp của phần mềm hệ thống. Phiên bản 8.2 của hệ điều hành EMUI có ngày phát hành cuối cùng, và phiên bản 8.2 của hệ điều hành EMUI có ngày phát hành sớm nhất. Phiên bản 8.1 của hệ điều hành EMUI được sử dụng trên thiết bị đầu cuối di động. Sau khi nhận thông tin phiên bản hiện tại được gửi bởi thiết bị đầu cuối di động, máy chủ OTA có thể xác định rằng phiên bản 8.1 của hệ điều hành EMUI được sử dụng trên thiết bị đầu cuối di động không là phiên bản cuối cùng. Máy chủ OTA có thể thực hiện phương pháp nâng cấp qua vô tuyến theo phương án này của sáng chế bằng cách sử dụng phiên bản mới của dữ liệu được mã hóa (cụ thể là, dữ liệu được mã hóa của phiên bản 8.2 của hệ điều hành EMUI) và phiên bản cũ của dữ liệu được mã hóa (dữ liệu được mã hóa của phiên bản 8.1 của hệ điều hành EMUI).

Phiên bản cũ của dữ liệu được mã hóa có thể nói đến dữ liệu thu được sau khi mã nguồn tương ứng với phiên bản cũ của phần mềm hệ thống (ví dụ, phiên bản 8.1 của hệ điều hành EMUI) được mã hóa và ký. Phiên bản mới của dữ liệu được mã hóa có thể nói đến dữ liệu thu được sau khi mã nguồn tương ứng với phiên bản mới của phần mềm hệ thống (ví dụ, phiên bản 8.2 của hệ điều hành EMUI) được mã hóa và ký. Mã nguồn tương ứng với phiên bản cũ của phần mềm hệ thống (ví dụ, phiên bản 8.1 của hệ điều hành EMUI) có thể được gọi là phiên bản cũ của dữ liệu ban đầu, và mã nguồn tương ứng với phiên bản mới của phần mềm hệ thống (ví dụ, phiên bản 8.2 của hệ điều hành EMUI) có thể được gọi là phiên bản mới của dữ liệu ban đầu.

Tham chiếu đến FIG.6A và FIG.6B, FIG.6A và FIG.6B là lưu đồ giản lược của phương pháp nâng cấp qua vô tuyến dựa trên kiến trúc hệ thống trên FIG.5a theo một

phương án thực hiện của sáng chế. Theo một phương án, khi quy trình của phương pháp nâng cấp qua vô tuyến theo phương án thực hiện sáng chế được thể hiện trên FIG.6A và FIG.6B được áp dụng cho kiến trúc hệ thống được thể hiện trên FIG.5b, bước 601 đến bước 603 trên FIG.6A và FIG.6B có thể được thực hiện bởi thiết bị đầu cuối. Sau khi bước 603 được thực hiện, thiết bị đầu cuối có thể tải lên dữ liệu OTA cho máy chủ OTA, và sau đó thiết bị đầu cuối di động có thể tải xuống dữ liệu OTA từ máy chủ OTA. Máy chủ OTA là máy chủ OTA 300 trên FIG.3, và thiết bị đầu cuối di động là thiết bị đầu cuối di động 200 trên FIG.2. Như được thể hiện trên FIG.6A và FIG.6B, phương pháp bao gồm các bước sau đây.

601. Máy chủ OTA loại bỏ dữ liệu chữ ký trong phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa, và giải mã phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa để thu được phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu.

Như được thể hiện trên FIG.7a, phiên bản cũ của dữ liệu được mã hóa bao gồm phiên bản cũ của dữ liệu ban đầu được mã hóa và phiên bản cũ của dữ liệu chữ ký. Phiên bản cũ của dữ liệu chữ ký có thể bao gồm bản tóm lược thông điệp (message digest) của phiên bản cũ của dữ liệu ban đầu được mã hóa, và thuật toán tóm lược thông điệp (message digest algorithm) (ví dụ, thuật toán hàm băm bảo mật 256 (secure hash algorithm 256, SHA 256)) để tạo ra bản tóm lược thông điệp. Máy chủ OTA có thể tính toán bản tóm lược thông điệp kiểm tra của phiên bản cũ của dữ liệu ban đầu được mã hóa bằng cách sử dụng thuật toán tóm lược thông điệp (ví dụ, SHA 256). Máy chủ OTA có thể xác định xem bản tóm lược thông điệp kiểm tra có giống với bản tóm lược thông điệp trong phiên bản cũ của dữ liệu chữ ký hay không. Nếu bản tóm lược thông điệp kiểm tra giống với bản tóm lược thông điệp trong phiên bản cũ của dữ liệu chữ ký, máy chủ OTA có thể xác định rằng phiên bản cũ của dữ liệu ban đầu không được điều chỉnh và là dữ liệu hợp lệ. Nói theo cách khác, phiên bản cũ của dữ liệu ban đầu có thể được sử dụng để tạo ra dữ liệu sai phân. Nếu bản tóm lược thông điệp kiểm tra khác với bản tóm lược thông điệp trong phiên bản cũ của dữ liệu chữ ký, máy chủ OTA có thể xác định rằng phiên bản cũ của dữ liệu ban đầu được điều chỉnh và là dữ liệu không hợp lệ, và máy chủ OTA có thể xóa bỏ dữ liệu không hợp lệ, và yêu cầu lấy lại phiên bản cũ của dữ liệu được mã hóa từ thiết bị đầu cuối của nhà cung

cấp của phần mềm hệ thống cho đến khi bản tóm lược thông điệp kiểm tra giống với bản tóm lược thông điệp trong dữ liệu chữ ký của phiên bản cũ.

Cách mã hóa mà được sử dụng bởi phiên bản cũ của dữ liệu ban đầu được mã hóa có thể là cách mã hóa đối xứng (ví dụ, thuật toán tiêu chuẩn mã hóa dữ liệu (data encryption standard, DES), thuật toán tiêu chuẩn mã hóa tiên tiến (advanced encryption standard, AES), hoặc thuật toán tiêu chuẩn mã hóa dữ liệu nhân ba (triple data encryption standard, 3DES)). Nói cách khác, khóa giống nhau (secret key: khóa bí mật) được sử dụng trong hoạt động mã hóa (encryption) và hoạt động giải mã (decryption).

Máy chủ OTA có thể giải mã phiên bản cũ của dữ liệu được mã hóa bằng cách sử dụng khóa thứ nhất được sử dụng trong hoạt động mã hóa của phiên bản cũ của dữ liệu được mã hóa, để thu được phiên bản cũ của dữ liệu ban đầu. Khóa thứ nhất có thể được lưu trữ trên máy chủ OTA, hoặc có thể được lưu trữ trên máy chủ lưu trữ khóa khác. Máy chủ lưu trữ khóa có thể không chỉ lưu trữ khóa của cách mã hóa đối xứng của máy chủ OTA, mà còn lưu trữ cặp khóa riêng tư và khóa công cộng của cách mã hóa bất đối xứng của máy chủ OTA, và các khóa khác cho hoạt động mã hóa hoặc giải mã dữ liệu.

Như được thể hiện trên FIG.7b, phiên bản mới của dữ liệu được mã hóa có thể bao gồm phiên bản mới của dữ liệu ban đầu được mã hóa và phiên bản mới của dữ liệu chữ ký. Phiên bản mới của dữ liệu chữ ký có thể bao gồm bản tóm lược thông điệp (message digest) của phiên bản mới của dữ liệu ban đầu được mã hóa mà được mã hóa bằng cách sử dụng khóa riêng tư, và thuật toán tóm lược thông điệp (ví dụ, SHA 256) để tạo ra bản tóm lược thông điệp. Máy chủ OTA có thể thu được bản tóm lược thông điệp của phiên bản mới của dữ liệu ban đầu được mã hóa thông qua hoạt động giải mã bằng cách sử dụng khóa công cộng khớp với khóa riêng tư. Máy chủ OTA có thể tính toán bản tóm lược thông điệp kiểm tra của phiên bản mới của dữ liệu ban đầu được mã hóa bằng cách sử dụng thuật toán tóm lược thông điệp (ví dụ, SHA 256). Nếu bản tóm lược thông điệp kiểm tra giống với bản tóm lược thông điệp trong phiên bản mới của dữ liệu chữ ký, máy chủ OTA có thể xác định rằng phiên bản mới của dữ liệu ban đầu không được điều chỉnh và là dữ liệu hợp lệ. Nói cách khác, phiên bản mới của dữ liệu ban đầu có thể được sử dụng để tạo ra dữ liệu sai phân. Nếu bản

tóm lược thông điệp kiểm tra khác với bản tóm lược thông điệp trong phiên bản mới của dữ liệu chữ ký, máy chủ OTA có thể xác định rằng phiên bản mới của dữ liệu ban đầu được điều chỉnh và là dữ liệu không hợp lệ, và máy chủ OTA xóa dữ liệu không hợp lệ, và yêu cầu lấy lại phiên bản mới của dữ liệu được mã hóa từ thiết bị đầu cuối của nhà cung cấp của phần mềm hệ thống cho đến khi bản tóm lược thông điệp kiểm tra giống với bản tóm lược thông điệp trong phiên bản mới của dữ liệu chữ ký.

Cách mã hóa mà được sử dụng bởi phiên bản mới của dữ liệu ban đầu được mã hóa có thể là cách mã hóa đối xứng (ví dụ, thuật toán DES, thuật toán AES, hoặc thuật toán 3DES). Nói theo cách khác, khóa giống nhau được sử dụng trong hoạt động mã hóa và giải mã của phiên bản mới của dữ liệu được mã hóa.

Máy chủ OTA có thể tách biệt phiên bản cũ của dữ liệu ban đầu được mã hóa và phiên bản mới của dữ liệu chữ ký trong phiên bản mới của dữ liệu được mã hóa, và máy chủ OTA có thể giải mã phiên bản mới của dữ liệu được mã hóa bằng cách sử dụng khóa thứ hai được sử dụng khi phiên bản mới của dữ liệu được mã hóa được mã hóa, để thu được phiên bản mới của dữ liệu ban đầu. Khóa thứ hai có thể được lưu trữ trên máy chủ OTA, hoặc có thể được lưu trữ trên máy chủ lưu trữ khóa. Khóa thứ hai có thể giống hoặc khác với khóa thứ nhất.

Theo một phương án, tuần tự để trước tiên loại bỏ dữ liệu chữ ký và sau đó giải mã dữ liệu chữ ký được thể hiện trên FIG.7a và FIG.7b không bị giới hạn, và trước tiên máy chủ OTA có thể giải mã phiên bản cũ của dữ liệu ban đầu được mã hóa trong phiên bản cũ của dữ liệu được mã hóa, để thu được phiên bản cũ của dữ liệu ban đầu và phiên bản cũ của dữ liệu chữ ký, và sau đó loại bỏ phiên bản cũ của dữ liệu chữ ký. Trước tiên máy chủ OTA có thể giải mã phiên bản mới của dữ liệu được mã hóa để thu được phiên bản mới của dữ liệu ban đầu và phiên bản mới của dữ liệu chữ ký, và sau đó loại bỏ phiên bản mới của dữ liệu chữ ký.

602. Máy chủ OTA tạo ra dữ liệu sai phân bằng cách sử dụng phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu.

Như được thể hiện trên FIG.8, máy chủ OTA có thể sử dụng phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu làm các đầu vào, để thực hiện phép toán sai phân bằng cách sử dụng thuật toán sai phân (ví dụ, thuật toán bsdiff hoặc thuật toán imgdiff), và xuất ra dữ liệu sai phân. Cách sai phân trong đó máy chủ OTA

thực hiện phép sai phân bằng cách sử dụng phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu có thể là phép sai phân dựa trên tệp, hoặc có thể là phép sai phân dựa trên khối dữ liệu (data block).

Sử dụng phép sai phân dựa trên tệp như một ví dụ, dữ liệu ban đầu có thể bao gồm một hoặc nhiều tệp dữ liệu, như file_1, file_2, và file_3, và phiên bản mới của dữ liệu có thể bao gồm một hoặc nhiều tệp dữ liệu, như file_1, file_2, file_3, và file_4. Thiết bị đầu cuối thứ nhất thực hiện, bằng cách sử dụng một hoặc nhiều tệp dữ liệu trong dữ liệu ban đầu và một hoặc nhiều tệp dữ liệu trong phiên bản mới của dữ liệu, phép toán sai phân bằng cách sử dụng thuật toán sai phân (ví dụ, thuật toán bsdiff hoặc thuật toán imgdiff), để thu được tệp sai phân (ví dụ, file_4) trong phiên bản mới của dữ liệu, và xác định dữ liệu chứa trong tệp sai phân (ví dụ, file_4) trong phiên bản mới của dữ liệu là dữ liệu sai phân.

Sử dụng phép sai phân dựa trên khối làm một ví dụ, phiên bản cũ của dữ liệu ban đầu có thể là gương (mirroring) (ví dụ, tệp ở định dạng ".img"). Gương của phiên bản cũ của dữ liệu ban đầu có thể được chia thành số lượng cụ thể (ví dụ, 32768) của các khối dữ liệu (data block) có kích thước cố định (ví dụ, 4 Kbyte) ở lớp lưu trữ cuối cùng của hệ thống tệp (file system). Phiên bản mới của dữ liệu ban đầu có thể là gương (ví dụ, tệp ở định dạng ".img"). Gương của phiên bản mới của dữ liệu có thể được chia thành số lượng cụ thể (ví dụ, 32768) của các khối dữ liệu (data block) có kích thước cố định (ví dụ, 4 Kbyte) ở lớp lưu trữ cuối cùng của hệ thống tệp. Ví dụ chỉ được sử dụng để giải thích phương án thực hiện này của sáng chế, và không nên hiểu là sự giới hạn. Máy chủ OTA có thể so sánh nhiều khối dữ liệu trong phiên bản cũ của dữ liệu ban đầu với nhiều khối dữ liệu trong phiên bản mới của dữ liệu ban đầu, và thực hiện phép toán sai phân trên nhiều khối dữ liệu trong dữ liệu ban đầu và nhiều khối dữ liệu trong phiên bản mới của dữ liệu, xác định một hoặc nhiều khối dữ liệu sai phân mà nằm trong nhiều khối dữ liệu trong phiên bản mới của dữ liệu ban đầu và khác với nhiều khối dữ liệu trong phiên bản cũ của dữ liệu ban đầu, và xác định dữ liệu chứa trong một hoặc nhiều khối dữ liệu sai phân là dữ liệu sai phân.

Ví dụ, nhiều khối dữ liệu trong phiên bản mới của dữ liệu ban đầu có thể là các khối dữ liệu từ khối 2_0 đến khối 2_32767, và nhiều khối dữ liệu trong dữ liệu ban đầu có thể là các khối dữ liệu từ khối 1_0 đến khối 1_32767. Các khối dữ liệu sai phân

khác với nhiều khối dữ liệu trong phiên bản cũ của dữ liệu ban đầu và mà nằm trong nhiều khối dữ liệu trong phiên bản mới của dữ liệu ban đầu có thể là các khối dữ liệu gồm khối 2_27001, khối 2_27002, khối 2_27003, và khối 2_22704. Thiết bị đầu cuối thứ nhất có thể xác định dữ liệu chứa trong các khối dữ liệu gồm khối 2_27001, khối 2_27002, khối 2_27003, và khối 2_27004 là dữ liệu sai phân.

603. Máy chủ OTA mã hóa và ký dữ liệu sai phân, và tạo ra thông tin tiêu đề, để thu được dữ liệu OTA.

Như được thể hiện trên FIG.9, máy chủ OTA có thể ở vị trí cố định của dữ liệu sai phân (ví dụ, ở phía trước của dữ liệu sai phân, trong dữ liệu sai phân, hoặc ở cuối của dữ liệu sai phân), và chèn một hoặc nhiều giá trị ngẫu nhiên mã hóa để tạo ra dữ liệu plaintext sai phân. Giá trị ngẫu nhiên mã hóa có thể được tạo ra bởi bộ tạo số ngẫu nhiên (random number generator) trên máy chủ OTA, hoặc có thể được tạo ra theo cách khác. Điều này không bị giới hạn ở đây.

Máy chủ OTA có thể mã hóa, bằng cách sử dụng khóa thứ ba, và thuật toán mã hóa đối xứng (ví dụ, thuật toán DES, thuật toán AES, thuật toán 3DES, hoặc thuật toán IDEA), dữ liệu sai phân và dữ liệu plaintext sai phân của một hoặc nhiều giá trị ngẫu nhiên mã hóa được chèn vào trong dữ liệu sai phân, để thu được dữ liệu ciphertext sai phân.

Máy chủ OTA có thể ký dữ liệu ciphertext sai phân bằng cách sử dụng khóa riêng tư của máy chủ OTA, thuật toán mã hóa bất đối xứng (ví dụ, thuật toán RSA), và thuật toán mã hóa EIGamal (EIGamal encryption algorithm). Khóa riêng tư của máy chủ OTA có thể được lưu trữ cục bộ trên máy chủ OTA, hoặc có thể được lưu trữ trên máy chủ lưu trữ khóa. Chữ ký dạng số cần được xác minh khi thiết bị đầu cuối di động nhận dữ liệu OTA, để xác định tính hợp lệ của dữ liệu OTA. Do đó, khóa công cộng khớp với khóa riêng tư của máy chủ OTA được lưu trữ trên thiết bị đầu cuối di động.

Thuật toán chữ ký dạng số RSA được sử dụng như một ví dụ. Trước tiên, máy chủ OTA tính toán bản tóm lược thông điệp của dữ liệu được mã hóa sai phân bằng cách sử dụng thuật toán tóm lược (ví dụ, thuật toán tóm lược thông điệp 2 (message digest algorithm 2, MD 2), thuật toán tóm lược thông điệp 4 (message digest algorithm 4, MD 4), thuật toán tóm lược thông điệp 5 (message digest algorithm 5, MD 5), thuật

toán hàm băm bảo mật 1 (secure hash algorithm 1, SHA 1), thuật toán hàm băm bảo mật 2 (secure hash algorithm 2, thuật toán SHA 2), hoặc thuật toán hàm băm bảo mật 256 (secure hash algorithm 256, SHA 256)). Sau đó, máy chủ OTA mã hóa bản tóm lược thông điệp bằng cách sử dụng khóa riêng tư của máy chủ OTA, và thuật toán RSA, để thu được chuỗi ký tự mà thu được sau khi bản tóm lược thông điệp được mã hóa, cụ thể là, để thu được kết quả chữ ký dạng số của dữ liệu được mã hóa sai phân.

Thông tin tiêu đề có thể bao gồm thông tin tiêu đề mã hóa dữ liệu sai phân và thông tin tiêu đề mã hóa dữ liệu đích. Thông tin tiêu đề mã hóa dữ liệu sai phân có thể bao gồm thông tin như thông tin thuật toán mã hóa được sử dụng khi máy chủ OTA mã hóa dữ liệu plaintext sai phân (ví dụ, thuật toán DES, thuật toán AES, hoặc thuật toán 3DES), và một hoặc nhiều giá trị ngẫu nhiên mã hóa sử dụng bởi máy chủ OTA trước khi mã hóa. Khóa thứ ba có thể là được lưu trữ cục bộ trên máy chủ OTA, hoặc có thể được lưu trữ trên máy chủ lưu trữ khóa. Khóa thứ ba có thể giống với khóa thứ nhất và khóa thứ hai, hoặc có thể khác với khóa thứ nhất và khóa thứ hai. Để giải mã dữ liệu OTA để thu được dữ liệu sai phân sau khi thiết bị đầu cuối di động nhận dữ liệu OTA, thiết bị đầu cuối di động cũng lưu trữ khóa thứ ba.

Máy chủ OTA có thể tạo ra thông tin tiêu đề mã hóa dữ liệu đích cho dữ liệu được mã hóa sai phân. Thông tin tiêu đề mã hóa dữ liệu đích có thể bao gồm thông tin xác minh chữ ký (bao gồm giá trị ngẫu nhiên kiểm tra), thông tin thuật toán tóm lược thông điệp (ví dụ, SHA 256), và thông tin thuật toán mã hóa bất đối xứng (ví dụ, thuật toán RSA) được sử dụng khi bản tóm lược thông điệp của dữ liệu được mã hóa sai phân được mã hóa. Như được thể hiện trên FIG.6A và FIG.6B, dữ liệu OTA có thể bao gồm dữ liệu được mã hóa sai phân, thông tin tiêu đề mã hóa dữ liệu sai phân, và thông tin tiêu đề mã hóa dữ liệu đích. Thông tin tiêu đề mã hóa dữ liệu đích có thể được sử dụng bởi thiết bị đầu cuối di động để xác minh dữ liệu chữ ký sai phân trước khi thiết bị đầu cuối di động hồi phục phiên bản mới của dữ liệu ban đầu bằng cách sử dụng dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu. Nếu việc xác minh thành công, nó chỉ báo rằng dữ liệu được mã hóa sai phân không được điều chỉnh, và thiết bị đầu cuối di động có thể hồi phục phiên bản mới của dữ liệu ban đầu. Thông tin tiêu đề mã hóa dữ liệu sai phân có thể được sử dụng bởi thiết bị đầu cuối di động để giải mã dữ liệu được mã hóa sai phân để thu được dữ liệu sai phân. Dữ liệu sai phân có thể được sử

dụng bởi thiết bị đầu cuối di động để tích hợp phiên bản mới của dữ liệu.

604. Thiết bị đầu cuối di động tải xuống dữ liệu OTA từ máy chủ OTA.

Máy chủ OTA có thể truyền thông với thiết bị đầu cuối di động bằng cách sử dụng giao thức truyền tải siêu văn bản trên lớp khe cắm bảo mật (hypertext transfer protocol over secure socket layer, HTTPS). Ví dụ, máy chủ OTA và thiết bị đầu cuối di động có thể tạo ra kênh bảo mật bằng cách sử dụng giao thức lớp khe cắm bảo mật (secure sockets layer, SSL) và/hoặc giao thức bảo mật lớp truyền tải (transport layer security, TLS), để thực hiện hoạt động truyền dữ liệu bảo mật.

605. Ứng dụng cập nhật trích xuất dữ liệu được mã hóa sai phân, và thông tin chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân từ dữ liệu OTA.

Do dữ liệu OTA bao gồm dữ liệu được mã hóa sai phân, thông tin tiêu đề mã hóa dữ liệu sai phân, và thông tin tiêu đề mã hóa dữ liệu đích, ứng dụng cập nhật có thể trích xuất thông tin mã hóa từ thông tin tiêu đề mã hóa dữ liệu sai phân, và ứng dụng cập nhật có thể trích xuất thông tin chữ ký từ thông tin tiêu đề mã hóa dữ liệu đích. Thông tin mã hóa bao gồm thông tin như thông tin thuật toán mã hóa của dữ liệu được mã hóa sai phân (ví dụ, thuật toán DES, thuật toán AES, hoặc thuật toán 3DES) và một hoặc nhiều giá trị ngẫu nhiên mã hóa của dữ liệu được mã hóa sai phân. Thông tin chữ ký bao gồm thông tin xác minh chữ ký (bao gồm giá trị ngẫu nhiên kiểm tra) của dữ liệu được mã hóa sai phân, thông tin thuật toán tóm lược thông điệp (ví dụ, SHA 256), và thông tin thuật toán mã hóa bất đối xứng (ví dụ, thuật toán RSA) dùng cho chữ ký.

606. Ứng dụng cập nhật bắt đầu CA thứ nhất, và gửi dữ liệu được mã hóa sai phân, thông tin chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân, và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống tới CA thứ nhất.

Cả ứng dụng cập nhật và CA thứ nhất là các ứng dụng khách (client application, CA) trong hệ điều hành môi trường thực thi phong phú (rich execution environment operating system, REE OS) của thiết bị đầu cuối di động. Phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống trên thiết bị đầu cuối di động có thể là dữ liệu thu được sau khi mã nguồn tương ứng với phiên bản cũ của phần mềm hệ thống trên thiết bị đầu cuối di động (ví dụ, phiên bản 8.1 của hệ điều hành EMUI) được mã hóa và ký bởi thiết bị đầu cuối di động. Mã nguồn tương ứng với phiên bản cũ của phần

mềm hệ thống (ví dụ, phiên bản 8.1 của hệ điều hành EMUI) chứa trong phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống giống với mã nguồn tương ứng với phiên bản cũ của phần mềm hệ thống chứa trong phiên bản cũ của dữ liệu được mã hóa. Mã nguồn tương ứng với phiên bản cũ của phần mềm hệ thống (ví dụ, phiên bản 8.1 của hệ điều hành EMUI) có thể được gọi là phiên bản cũ của dữ liệu ban đầu, và mã nguồn tương ứng với phiên bản mới của phần mềm hệ thống (ví dụ, phiên bản 8.2 của hệ điều hành EMUI) có thể được gọi là phiên bản mới của dữ liệu ban đầu.

Như được thể hiện trên FIG.10A và FIG.10B, dữ liệu OTA được tải xuống bởi thiết bị đầu cuối di động từ máy chủ OTA có thể bao gồm thông tin tiêu đề và dữ liệu được mã hóa sai phân. Thông tin tiêu đề bao gồm thông tin tiêu đề mã hóa dữ liệu đích và thông tin tiêu đề mã hóa dữ liệu sai phân. Thông tin tiêu đề mã hóa dữ liệu đích bao gồm dữ liệu chữ ký của dữ liệu được mã hóa sai phân, và thông tin tiêu đề mã hóa dữ liệu sai phân bao gồm thông tin mã hóa của dữ liệu được mã hóa sai phân. Ứng dụng cập nhật có thể nhận dạng, dựa trên thông tin tiêu đề của dữ liệu OTA, rằng dữ liệu OTA là dữ liệu được mã hóa.

607. CA thứ nhất chuyển tiếp dữ liệu được mã hóa sai phân, thông tin chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân, và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống tới ứng dụng tin cậy (trusted application, TA) trong hệ điều hành môi trường thực thi tin cậy (trusted execution environment operating system, TEE OS).

608. TA thực hiện việc xác minh chữ ký và việc giải mã trên dữ liệu được mã hóa sai phân và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống, để thu được dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu.

Như được thể hiện trên FIG.10A và FIG.10B, sau khi TA nhận dữ liệu được mã hóa sai phân, thông tin chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân, và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống được chuyển tiếp bởi CA thứ nhất, TA có thể xác thực chữ ký của dữ liệu được mã hóa sai phân. Ví dụ, thiết bị đầu cuối di động lưu trữ cục bộ khóa công cộng của máy chủ OTA. TA có thể giải mã dữ liệu chữ ký sai phân bằng cách sử dụng khóa công cộng của máy chủ OTA để thu được bản tóm lược thông điệp của dữ liệu ciphertext sai phân. TA có thể tính toán bản tóm lược thông điệp kiểm tra của dữ liệu ciphertext sai phân bằng cách sử

dụng thuật toán tóm lược thông điệp (ví dụ, SHA 256). Nếu bản tóm lược thông điệp kiểm tra giống với bản tóm lược thông điệp thu được thông qua hoạt động giải mã bằng cách sử dụng khóa công cộng, thì dữ liệu ciphertext sai phân không được điều chỉnh, và việc xác minh chữ ký thành công. TA có thể sử dụng dữ liệu ciphertext sai phân để tích hợp phiên bản mới của dữ liệu ban đầu. Nếu bản tóm lược thông điệp kiểm tra khác với bản tóm lược thông điệp thu được thông qua hoạt động giải mã bằng cách sử dụng khóa công cộng, dữ liệu ciphertext sai phân được điều chỉnh, và việc xác minh chữ ký thất bại. Thiết bị đầu cuối di động tải xuống lại dữ liệu OTA từ máy chủ OTA cho đến khi việc xác minh của dữ liệu chữ ký sai phân trong dữ liệu được mã hóa sai phân thành công.

Theo một phương án, trước khi thiết bị đầu cuối di động tải xuống dữ liệu OTA từ máy chủ OTA, phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống được lưu trữ trên thiết bị đầu cuối di động là dữ liệu thu được sau khi TA mã hóa phiên bản cũ của dữ liệu ban đầu và ký phiên bản cũ của dữ liệu ban đầu bằng cách sử dụng khóa riêng tư của thiết bị đầu cuối di động. Sau khi nhận dữ liệu được mã hóa sai phân, thông tin chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân, và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống được chuyển tiếp bởi CA thứ nhất, trước tiên TA có thể xác minh chữ ký của phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống bằng cách sử dụng khóa công cộng của thiết bị đầu cuối di động. Sau khi việc xác minh chữ ký thành công, TA giải mã phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống để thu được phiên bản cũ của dữ liệu ban đầu. Ví dụ, TA có thể thu được bản tóm lược thông điệp của phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống bằng cách sử dụng khóa công cộng của thiết bị đầu cuối di động và thuật toán giải mã bất đối xứng (ví dụ, thuật toán RSA). Sau đó, TA có thể tính toán lại bản tóm lược thông điệp kiểm tra của phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống bằng cách sử dụng thuật toán tóm lược thông điệp (ví dụ, SHA 256) dùng cho chữ ký. Nếu bản tóm lược thông điệp kiểm tra của phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống giống với bản tóm lược thông điệp của phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống thu được thông qua hoạt động giải mã bằng cách sử dụng khóa công cộng, dữ liệu được mã hóa sai phân không được điều chỉnh, và việc xác minh chữ ký thành công.

Sau khi việc xác minh chữ ký của TA trên dữ liệu được mã hóa sai phân và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống thành công, TA có thể giải mã dữ liệu được mã hóa sai phân và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống, để thu được dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu.

Ví dụ, TA giải mã dữ liệu sai phân. TA có thể giải mã dữ liệu được mã hóa sai phân bằng cách sử dụng khóa thứ ba được sử dụng khi máy chủ OTA mã hóa dữ liệu sai phân, thuật toán mã hóa (ví dụ, thuật toán DES) trong thông tin mã hóa, và giá trị ngẫu nhiên, để thu được dữ liệu sai phân. Khóa thứ ba có thể được định trước trong thiết bị đầu cuối di động, hoặc có thể được gửi bởi máy chủ OTA bằng cách sử dụng kênh truyền bảo mật. Điều này không bị giới hạn ở đây.

Ví dụ, TA giải mã phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống. TA có thể giải mã phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống bằng cách sử dụng khóa thứ tư và thuật toán mã hóa và giải mã định trước (ví dụ, thuật toán AES), để thu được phiên bản cũ của dữ liệu ban đầu. Khóa thứ tư được định trước trong thiết bị đầu cuối di động, và khóa thứ tư có thể giống hoặc khác với khóa thứ ba. Theo một phương án, khi TA mã hóa hoặc giải mã dữ liệu được lưu trữ cục bộ trong thiết bị đầu cuối di động, TA có thể thực hiện việc mã hóa chip hoặc việc giải mã chip. Đối với phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống, TA có thể giải mã phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống theo cách mã hóa/giải mã chip, để thu được phiên bản cũ của dữ liệu ban đầu.

609. TA tích hợp và hồi phục phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu.

TA có thể thực hiện việc tích hợp dữ liệu bằng cách sử dụng thuật toán hồi phục sai phân (ví dụ, thuật toán bspatch) dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu, để hồi phục phiên bản mới của dữ liệu ban đầu. Phiên bản mới của dữ liệu ban đầu là dữ liệu hoàn thành thu được sau khi phần mềm hệ thống được cập nhật.

610. TA mã hóa và ký phiên bản mới của dữ liệu ban đầu để thu được phiên bản mới của dữ liệu được mã hóa.

Như được thể hiện trên FIG.10A và FIG.10B, TA có thể mã hóa phiên bản mới của dữ liệu ban đầu. TA có thể mã hóa phiên bản mới của dữ liệu ban đầu bằng cách

sử dụng khóa thứ năm chứa trong TEE, giá trị ngẫu nhiên (ví dụ, mật khẩu khởi động của thiết bị đầu cuối di động) chứa trong TEE, và thuật toán mã hóa định trước (ví dụ, thuật toán AES). Theo một phương án, TA có thể còn mã hóa phiên bản mới của dữ liệu ban đầu theo cách chip mã hóa/giải mã, để tạo ra phiên bản mới của dữ liệu ban đầu được mã hóa. Ví dụ, khóa thứ năm có thể là khóa lưu trữ bảo mật (secure storage key, SSK). Ví dụ, các giá trị của các SSK trong các thiết bị đầu cuối di động khác nhau là khác nhau. Khi TEE bắt đầu, bộ nhận dạng chip (chip ID) và khóa phần cứng duy nhất (khóa phần cứng duy nhất, HUK) được sử dụng để thu được giá trị của SSK thông qua sự tính toán bằng cách sử dụng mã xác thực thông điệp dựa trên hàm băm (hash-based message authentication code, HMAC). Cả HUK và chip ID được định trước trong chip của thiết bị đầu cuối di động, hoặc có thể là các khóa khác như khóa lưu trữ ứng dụng tin cậy (trusted application storage key, TASK). Điều này không bị giới hạn cụ thể ở đây.

Theo một phương án, sau khi TA mã hóa phiên bản mới của dữ liệu ban đầu để thu được phiên bản mới của dữ liệu ban đầu được mã hóa, TA có thể còn ký phiên bản mới của dữ liệu ban đầu được mã hóa, để thu được phiên bản mới của dữ liệu được mã hóa. TA có thể ký phiên bản mới của dữ liệu ban đầu được mã hóa bằng cách sử dụng khóa riêng tư của thiết bị đầu cuối di động và thuật toán mã hóa bất đối xứng (ví dụ, thuật toán RSA, thuật toán mã hóa EIGamal, và thuật toán DSA). Theo cách thay thế, khóa được sử dụng bởi TA để ký phiên bản mới của dữ liệu ban đầu được mã hóa có thể là khóa lưu trữ bảo mật (SSK). Ví dụ, các giá trị của các SSK trong các thiết bị đầu cuối di động khác nhau là khác nhau. Khi TEE bắt đầu, chip ID và khóa phần cứng duy nhất HUK được sử dụng để thu được giá trị của SSK thông qua sự tính toán bằng cách sử dụng mã xác thực thông điệp dựa trên hàm băm (HMAC). Cả HUK và chip ID được định trước trong chip của thiết bị đầu cuối di động, hoặc các khóa khác như khóa lưu trữ ứng dụng tin cậy (TASK). Điều này không bị giới hạn cụ thể ở đây.

Thuật toán chữ ký dạng số RSA được sử dụng như một ví dụ. Trước tiên, TA tính toán bản tóm lược thông điệp của phiên bản mới của dữ liệu ban đầu được mã hóa bằng cách sử dụng thuật toán tóm lược (ví dụ, MD 2, MD 4, MD 5, SHA 1, SHA 2, hoặc SHA 256). Sau đó, TA mã hóa bản tóm lược thông điệp bằng cách sử dụng khóa

riêng tư của thiết bị đầu cuối di động và thuật toán RSA, để thu được chuỗi ký tự mà thu được sau khi bản tóm lược thông điệp được mã hóa, cụ thể là, để thu được kết quả chữ ký dạng số của phiên bản mới của dữ liệu ban đầu được mã hóa. TA có thể tạo ra phiên bản mới của dữ liệu được mã hóa dựa trên kết quả chữ ký dạng số và phiên bản mới của dữ liệu ban đầu được mã hóa.

611. TA gửi phiên bản mới của dữ liệu được mã hóa tới CA thứ nhất. 612. CA thứ nhất chuyển tiếp phiên bản mới của dữ liệu được mã hóa tới ứng dụng cập nhật. 613. Sau khi nhận phiên bản mới của dữ liệu được mã hóa được gửi bởi CA thứ nhất, ứng dụng cập nhật có thể ghi phiên bản mới của dữ liệu được mã hóa vào trong phương tiện lưu trữ.

CA thứ nhất có thể chuyển tiếp phiên bản mới của dữ liệu được mã hóa tới ứng dụng cập nhật. Ứng dụng cập nhật có thể xóa bỏ phiên bản cũ của dữ liệu được mã hóa trong phân vùng lưu trữ tương ứng với phần mềm hệ thống, và ghi phiên bản mới của dữ liệu được mã hóa vào trong phương tiện lưu trữ của thiết bị đầu cuối di động (ví dụ, phân vùng hệ thống (system) của bộ nhớ chỉ đọc (read-only memory, ROM)) để hoàn thành việc cập nhật và nâng cấp của phần mềm hệ thống từ phiên bản cũ thành phiên bản mới.

Theo phương án này của sáng chế, phần mềm hệ thống được nâng cấp bằng cách sử dụng công nghệ qua vô tuyến (OTA). Khi tạo ra dữ liệu OTA, trước tiên máy chủ OTA có thể loại bỏ dữ liệu chữ ký của dữ liệu được mã hóa (cụ thể là, phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa) và giải mã dữ liệu được mã hóa. Sau đó, dữ liệu sai phân được tạo ra dựa trên phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu của phần mềm hệ thống mà thu được thông qua hoạt động giải mã, và dữ liệu OTA bao gồm dữ liệu sai phân được gửi tới thiết bị đầu cuối di động. Sau khi nhận dữ liệu OTA, thiết bị đầu cuối di động có thể hồi phục phiên bản mới của dữ liệu ban đầu dựa trên phiên bản cũ được lưu trữ cục bộ của dữ liệu ban đầu của phần mềm hệ thống và dữ liệu sai phân trong dữ liệu OTA, và mã hóa và ký phiên bản mới của dữ liệu ban đầu bằng cách sử dụng ứng dụng tin cậy (TA: trusted application) trong môi trường thực thi tin cậy (REE: trusted execution environment) của thiết bị đầu cuối di động. Theo cách này, do máy chủ OTA thực hiện phép sai phân chỉ trên phiên bản mới của dữ liệu ban đầu và phiên bản cũ

của dữ liệu ban đầu, dữ liệu chữ ký của phiên bản mới của dữ liệu ban đầu được loại bỏ khỏi dữ liệu sai phân, sao cho kích thước của dữ liệu được truyền bởi máy chủ OTA tới thiết bị đầu cuối di động cho việc nâng cấp phần mềm hệ thống được giảm, và việc nâng cấp nhanh của phần mềm hệ thống được thực hiện.

Theo một phương án, bước 609 để tích hợp và hồi phục phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu trên FIG.6A và FIG.6B có thể được thực hiện bởi CA thứ nhất.

Như được thể hiện trên FIG.11A và FIG.11B, ứng dụng cập nhật có thể gửi dữ liệu chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân, dữ liệu được mã hóa sai phân, và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống của thiết bị đầu cuối di động tới CA thứ nhất. CA thứ nhất có thể chuyển tiếp dữ liệu chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân, dữ liệu được mã hóa sai phân, và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống thiết bị đầu cuối di động tới TA. TA có thể xác thực chữ ký của dữ liệu được mã hóa sai phân và chữ ký của phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống. Sau khi việc xác minh chữ ký thành công, TA có thể giải mã dữ liệu được mã hóa sai phân để thu được dữ liệu sai phân, và giải mã phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống để thu được phiên bản cũ của dữ liệu ban đầu. TA có thể truyền dữ liệu sai phân và phiên bản cũ của dữ liệu tới CA thứ nhất.

CA thứ nhất có thể tích hợp và hồi phục phiên bản mới của dữ liệu ban đầu bằng cách sử dụng thuật toán hồi phục sai phân (ví dụ, thuật toán bspatch) dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu. Phiên bản mới của dữ liệu ban đầu là dữ liệu hoàn thành thu được sau khi phần mềm hệ thống được cập nhật. CA thứ nhất có thể gửi phiên bản mới của dữ liệu ban đầu tới TA.

TA có thể mã hóa phiên bản mới của dữ liệu ban đầu (ví dụ, bằng cách sử dụng tiêu chuẩn mã hóa tiên tiến (advanced encryption standard, AES) để thu được phiên bản mới của dữ liệu ban đầu được mã hóa. TA có thể ký phiên bản mới của dữ liệu ban đầu được mã hóa bằng cách sử dụng thuật toán mã hóa bất đối xứng (ví dụ, thuật toán RSA, thuật toán mã hóa EIGamal, và thuật toán DSA) để tạo ra phiên bản mới của dữ liệu được mã hóa. Phiên bản mới của dữ liệu được mã hóa bao gồm phiên bản mới của dữ liệu ban đầu được mã hóa và kết quả chữ ký của phiên bản mới của dữ liệu ban đầu

được mã hóa. TA có thể gửi phiên bản mới của dữ liệu được mã hóa tới CA thứ nhất.

CA thứ nhất có thể chuyển tiếp phiên bản mới của dữ liệu được mã hóa tới ứng dụng cập nhật. Ứng dụng cập nhật có thể ghi phiên bản mới của dữ liệu được mã hóa vào trong phương tiện lưu trữ của thiết bị đầu cuối di động (ví dụ, phân vùng hệ thống (system) của bộ nhớ chỉ đọc (read-only memory, ROM)), để hoàn thành việc cập nhật và nâng cấp của phần mềm hệ thống.

Theo phương án thực hiện này của sáng chế, quá trình hồi phục phiên bản mới của dữ liệu ban đầu bằng cách sử dụng thuật toán hồi phục sai phân (ví dụ, thuật toán bspatch) dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu được thực hiện bởi ứng dụng khách (CA) trong môi trường thực thi phong phú (REE) của thiết bị đầu cuối di động. CA có thể chia sẻ một phần của tác vụ xử lý dữ liệu của TA, sao cho tính bảo mật của dữ liệu trong việc cập nhật phần mềm hệ thống có thể được đảm bảo, và tốc độ của việc cập nhật phần mềm hệ thống trên thiết bị đầu cuối di động có thể được cải thiện.

Phần dưới đây mô tả phương án thực hiện của bước 601 đến bước 602 được thể hiện trên FIG.6A và FIG.6B theo phương án này của sáng chế.

Khi máy chủ OTA hoặc hệ thống tệp của thiết bị đầu cuối di động lưu trữ dữ liệu của phần mềm hệ thống, gương hệ thống (ví dụ, gương ở định dạng ext4) trước tiên có thể được tạo ra từ dữ liệu của phần mềm hệ thống, và sau đó, hệ thống tệp có thể chia gương hệ thống thành các khối dữ liệu (data block) có kích thước cố định (ví dụ, 4 Kbyte), và tạo ra cây hàm băm (hash tree) cho gương hệ thống.

Ví dụ, đối với quá trình tạo ra cây hàm băm cho gương hệ thống bởi hệ thống tệp, tham chiếu đến FIG.12. Như được thể hiện trên FIG.12, hệ thống tệp chia gương hệ thống thành các khối dữ liệu 4 Kbyte (ví dụ, các khối dữ liệu blk 0_0 đến blk 0_32767) ở lớp 0. Hệ thống tệp có thể chọn giá trị ngẫu nhiên kiểm tra (ví dụ, giá trị mã hóa hệ thập lục phân), và tính toán, bằng cách sử dụng thuật toán hàm băm (ví dụ, SHA 256), các giá trị hàm băm thu được sau khi tất cả các khối dữ liệu ở lớp 0 được bổ sung cho giá trị ngẫu nhiên kiểm tra. Các giá trị hàm băm này được kết hợp vào các khối dữ liệu 4 Kbyte để tạo ra lớp 1. Ví dụ, các khối dữ liệu ở lớp 1 có thể bao gồm các khối dữ liệu blk 1_0 đến blk 1_255. Tiếp theo, hệ thống tệp tính toán, bằng cách sử dụng thuật toán hàm băm giống nhau (ví dụ, SHA 256), các giá trị hàm băm

thu được sau khi tất cả các khối dữ liệu ở lớp 1 được bổ sung cho giá trị ngẫu nhiên kiểm tra, và fills khối dữ liệu (ví dụ, khối dữ liệu blk 2_0 đến khối dữ liệu blk 2_1) ở lớp 2 với giá trị hàm băm của mỗi khối dữ liệu ở lớp 1, cho đến khi các giá trị hàm băm của tất cả các khối dữ liệu trong lớp trước đó có thể được đưa vào trong một khối dữ liệu (ví dụ, khối dữ liệu blk_root). Trong trường hợp này, quá trình tạo ra cây hàm băm được hoàn thành. Khối dữ liệu cuối cùng là khối dữ liệu hàm băm gốc của cây hàm băm, và khối dữ liệu hàm băm gốc bao gồm giá trị hàm băm gốc (root hash) của cây hàm băm. Nếu khối dữ liệu ở lớp không thể được điền một cách chính xác bởi các giá trị hàm băm của lớp trước đó, "0" có thể được điền vào khối dữ liệu để thu được kích thước yêu cầu bằng 4 Kbyte. Để tạo ra cây hàm băm, các khối dữ liệu ở lớp 2 cần được kết hợp bên trên các khối dữ liệu ở lớp 1, các khối dữ liệu ở lớp 3 cần được kết hợp bên trên các khối dữ liệu ở lớp 2, v.v., cho đến khi lớp trên cùng là khối dữ liệu hàm băm gốc (ví dụ, blk_root). Hệ thống tệp có thể thực hiện chữ ký được mã hóa tren khối dữ liệu hàm băm gốc bằng cách sử dụng thuật toán mã hóa bất đối xứng (ví dụ, thuật toán RSA) và khóa riêng tư định trước, để thu được kết quả chữ ký. Hệ thống tệp có thể ghi khối dữ liệu chứa trong cây hàm băm vào trong phân vùng lưu trữ logic tương ứng (ví dụ, phân vùng hệ thống (system)) mà được lưu trữ trong phương tiện lưu trữ cùng với gương của phần mềm hệ thống.

Do đó, khi lưu trữ dữ liệu của phần mềm hệ thống, hệ thống tệp lưu trữ gương hệ thống (ví dụ, dữ liệu trong các khối dữ liệu ở lớp 0), cây hàm băm, kết quả chữ ký, và giá trị ngẫu nhiên kiểm tra cùng nhau vào trong phân vùng lưu trữ logic tương ứng (ví dụ, phân vùng hệ thống) trong phương tiện lưu trữ. Phân vùng lưu trữ logic có thể bao gồm ba loại của các khối dữ liệu: khối dữ liệu chỉ bằng 0, khối dữ liệu tệp, và khối dữ liệu không tệp không bằng 0. Khối dữ liệu chỉ bằng 0 có thể là khối dữ liệu mà nằm ở phân vùng lưu trữ logic và không được điền dữ liệu. Khối dữ liệu tệp có thể là khối dữ liệu tương ứng với gương hệ thống, ví dụ, khối dữ liệu ở lớp 0 được thể hiện trên FIG.12. Khối dữ liệu không tệp không bằng 0 có thể bao gồm các khối dữ liệu (ví dụ, các khối dữ liệu ở lớp 1 đến lớp 2 được thể hiện trên FIG.12) tương ứng với cây hàm băm và các khối dữ liệu tương ứng với kết quả chữ ký.

Khi máy chủ OTA tạo ra dữ liệu sai phân theo cách sai phân dựa trên khối (block), do gương hệ thống trong phiên bản cũ của dữ liệu được mã hóa khác với

gương hệ thống trong phiên bản mới của dữ liệu được mã hóa, khối dữ liệu hàm băm tương ứng với cây hàm băm trong phiên bản cũ của dữ liệu được mã hóa cũng khác với khối dữ liệu hàm băm tương ứng với cây hàm băm trong phiên bản mới của dữ liệu được mã hóa. Nếu máy chủ OTA thực hiện phép toán sai phân trên tất cả các khối dữ liệu không bằng 0 (bao gồm các khối dữ liệu tệp và các khối dữ liệu không tệp không bằng 0) trong phiên bản cũ của dữ liệu được mã hóa và tất cả các khối dữ liệu không bằng 0 (bao gồm các khối dữ liệu tệp và các khối dữ liệu không tệp không bằng 0) trong phiên bản mới của dữ liệu được mã hóa, khi máy chủ OTA tạo ra dữ liệu OTA, do khối dữ liệu không tệp không bằng 0 bao gồm khối dữ liệu hàm băm và khối dữ liệu kiểm tra, dữ liệu OTA bao gồm lượng dữ liệu trong khối dữ liệu hàm băm. Theo cách này, toàn bộ lượng dữ liệu của dữ liệu OTA được tăng lên, vốn không tạo điều kiện thuận lợi cho việc cập nhật phần mềm hệ thống trên thiết bị đầu cuối di động.

Do đó, phần dưới đây mô tả quá trình tạo ra dữ liệu OTA theo cách sai phân dựa trên khối (block) theo một phương án thực hiện của sáng chế. Khi máy chủ OTA tạo ra dữ liệu sai phân, các khối dữ liệu tương ứng với cây hàm băm được loại bỏ, và phiên bản mới của khối dữ liệu kiểm tra được giữ lại. Sau đó, khi hồi phục dữ liệu sai phân, thiết bị đầu cuối di động tạo ra, bằng cách sử dụng phiên bản mới được hồi phục của dữ liệu ban đầu và giá trị ngẫu nhiên kiểm tra trong phiên bản mới của khối dữ liệu kiểm tra, khối dữ liệu hàm băm tương ứng với phiên bản mới của dữ liệu ban đầu, và lưu trữ khối dữ liệu hàm băm vào trong phương tiện lưu trữ. Do khối dữ liệu hàm băm được loại bỏ khỏi dữ liệu OTA, và thay vào đó khối dữ liệu hàm băm được tạo ra trên thiết bị đầu cuối di động, lượng dữ liệu của dữ liệu cập nhật phần mềm hệ thống (dữ liệu OTA) được tải xuống bởi thiết bị đầu cuối di động được giảm.

Như được thể hiện trên FIG.13A và FIG.13B, máy chủ OTA có thể phân loại các khối dữ liệu theo phân vùng lưu trữ logic (ví dụ, phân vùng hệ thống) trong đó phiên bản cũ của dữ liệu ban đầu được đặt vào ba loại: khối dữ liệu chỉ bằng 0, phiên bản cũ của khối dữ liệu tệp, và phiên bản không cũ không bằng 0 của khối dữ liệu tệp. Phiên bản không cũ không bằng 0 của khối dữ liệu tệp có thể bao gồm phiên bản cũ của khối dữ liệu hàm băm và phiên bản cũ của khối dữ liệu kiểm tra. Phiên bản cũ của khối dữ liệu hàm băm được sử dụng để lưu trữ dữ liệu như cây hàm băm của phiên bản cũ của khối dữ liệu tệp. Phiên bản cũ của khối dữ liệu kiểm tra được sử dụng để

lưu trữ giá trị ngẫu nhiên kiểm tra mà được sử dụng để tạo ra cây hàm băm của phiên bản cũ của khối dữ liệu tệp, và kết quả chữ ký của phiên bản cũ của khối dữ liệu tệp (cụ thể là, dữ liệu thu được sau khi giá trị hàm băm gốc được mã hóa bằng cách sử dụng khóa riêng tư).

Các khối dữ liệu theo phân vùng lưu trữ logic (ví dụ, phân vùng hệ thống) trong đó phiên bản mới của dữ liệu ban đầu được định vị có thể được phân loại thành ba loại: khối dữ liệu chỉ bằng 0, phiên bản mới của khối dữ liệu tệp, và phiên bản không mới không bằng 0 của khối dữ liệu tệp. Phiên bản không mới không bằng 0 của khối dữ liệu tệp có thể bao gồm phiên bản mới của khối dữ liệu hàm băm và phiên bản mới của khối dữ liệu kiểm tra. Phiên bản mới của khối dữ liệu hàm băm được sử dụng để lưu trữ dữ liệu như cây hàm băm của phiên bản mới của khối dữ liệu tệp. Phiên bản mới của khối dữ liệu kiểm tra được sử dụng để lưu trữ giá trị ngẫu nhiên kiểm tra được sử dụng để tạo ra cây hàm băm của phiên bản mới của khối dữ liệu tệp, và kết quả chữ ký của phiên bản mới của khối dữ liệu tệp (cụ thể là, dữ liệu thu được sau khi giá trị hàm băm gốc được mã hóa bằng cách sử dụng khóa riêng tư).

Máy chủ OTA có thể loại bỏ phiên bản cũ của khối dữ liệu hàm băm khỏi phiên bản không cũ không bằng 0 của khối dữ liệu tệp, và giữ lại phiên bản cũ của khối dữ liệu kiểm tra. Máy chủ OTA có thể loại bỏ phiên bản mới của khối dữ liệu hàm băm khỏi phiên bản không mới không bằng 0 của khối dữ liệu tệp, và giữ lại phiên bản mới của khối dữ liệu kiểm tra.

Máy chủ OTA có thể thực hiện phép sai phân khối dữ liệu trên phiên bản mới của khối dữ liệu tệp, phiên bản mới của khối dữ liệu kiểm tra, phiên bản cũ của khối dữ liệu tệp, và phiên bản cũ của khối dữ liệu kiểm tra theo cách sai phân dựa trên khối dữ liệu, để tạo ra dữ liệu sai phân. Phép sai phân khối dữ liệu trên phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu được thực hiện dựa trên cách sai phân dựa trên khối dữ liệu, để tạo ra dữ liệu sai phân. Dữ liệu sai phân bao gồm khối dữ liệu mà khác với phiên bản cũ của dữ liệu ban đầu mà phiên bản cũ của khối dữ liệu hàm băm được loại bỏ khỏi đó và nằm trong phiên bản mới của dữ liệu ban đầu mà phiên bản mới của khối dữ liệu hàm băm được loại bỏ khỏi đó. Phiên bản mới của khối dữ liệu kiểm tra khác với phiên bản cũ của khối dữ liệu kiểm tra. Do đó, dữ liệu sai phân bao gồm khối dữ liệu mà khác với phiên bản cũ của khối dữ liệu tệp và nằm

trong phiên bản mới của khối dữ liệu tệp, và phiên bản mới của khối dữ liệu kiểm tra.

Máy chủ OTA có thể tạo ra dữ liệu OTA dựa trên dữ liệu sai phân, và gửi dữ liệu OTA tới thiết bị đầu cuối di động. Máy chủ OTA có thể gửi dữ liệu OTA tới thiết bị đầu cuối di động theo cách truyền thông không dây như truyền thông mạng Wi-Fi hoặc truyền thông trạm gốc.

Sau khi nhận dữ liệu OTA, thiết bị đầu cuối di động có thể phân tích dữ liệu sai phân từ dữ liệu OTA. Thiết bị đầu cuối di động có thể hồi phục, bằng cách sử dụng thuật toán hồi phục sai phân (ví dụ, thuật toán bspatch), phiên bản mới của tệp và khối dữ liệu và phiên bản mới của khối dữ liệu kiểm tra dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu được lưu trữ cục bộ trong thiết bị đầu cuối di động. Do phiên bản mới của khối dữ liệu kiểm tra bao gồm giá trị ngẫu nhiên kiểm tra được sử dụng để tạo ra cây hàm băm của phiên bản mới của khối dữ liệu tệp, thiết bị đầu cuối di động có thể tạo ra cây hàm băm của phiên bản mới của khối dữ liệu tệp dựa trên giá trị ngẫu nhiên kiểm tra và phiên bản mới của khối dữ liệu tệp, để thu được phiên bản mới của khối dữ liệu hàm băm. Đối với quá trình tạo ra phiên bản mới của khối dữ liệu tệp bởi thiết bị đầu cuối di động, tham chiếu đến phương án nêu trên được thể hiện trên FIG.12. Các chi tiết không được mô tả lại ở đây.

Sau khi tạo ra phiên bản mới của khối dữ liệu hàm băm, thiết bị đầu cuối di động có thể mã hóa phiên bản mới của dữ liệu ban đầu bao gồm phiên bản mới của khối dữ liệu tệp, phiên bản mới của khối dữ liệu hàm băm, và phiên bản mới của khối dữ liệu kiểm tra, và lưu trữ phiên bản mới của dữ liệu ban đầu vào trong tương ứng phân vùng lưu trữ logic (ví dụ, phân vùng hệ thống) trong phương tiện lưu trữ, để hoàn thành việc cập nhật và nâng cấp của phần mềm hệ thống từ phiên bản cũ thành phiên bản mới.

Tóm lại, các phương án nêu trên chỉ được dự tính để mô tả các giải pháp kỹ thuật của sáng chế, mà không để giới hạn sáng chế. Mặc dù sáng chế được mô tả chi tiết tham chiếu đến các phương án thực hiện nêu trên, nhưng những người có hiểu biết trung bình trong lĩnh vực kỹ thuật sẽ hiểu rằng chúng có thể vẫn có các điều chỉnh cho các giải pháp kỹ thuật được mô tả trong các phương án thực hiện nêu trên hoặc có thể tạo ra các thay thế tương đương cho một số dấu hiệu kỹ thuật của chúng, mà không tách rời khỏi phạm vi bảo hộ của các giải pháp kỹ thuật của các phương án thực hiện

của sáng chế.

YÊU CẦU BẢO HỘ

1. Phương pháp gửi dữ liệu bằng công nghệ qua vô tuyến (OTA: over-the-air), bao gồm các bước:

thu được, bởi máy chủ thứ nhất, phiên bản mới của dữ liệu được mã hóa và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống được áp dụng cho thiết bị đầu cuối di động;

giải mã, bởi máy chủ thứ nhất, phiên bản mới của dữ liệu được mã hóa để thu được phiên bản mới của dữ liệu ban đầu, và giải mã phiên bản cũ của dữ liệu được mã hóa để thu được phiên bản cũ của dữ liệu ban đầu;

thực hiện, bởi máy chủ thứ nhất, phép sai phân trên phiên bản mới của dữ liệu ban đầu và phiên bản cũ của dữ liệu ban đầu, để thu được dữ liệu sai phân, trong đó dữ liệu sai phân bao gồm dữ liệu mà nằm trong phiên bản mới của dữ liệu ban đầu và khác với phiên bản cũ của dữ liệu ban đầu;

tạo ra, bởi máy chủ thứ nhất, dữ liệu OTA, trong đó dữ liệu OTA bao gồm dữ liệu được mã hóa sai phân và thông tin tiêu đề, dữ liệu được mã hóa sai phân là dữ liệu thu được sau khi dữ liệu sai phân được mã hóa hoặc dữ liệu được mã hóa sai phân là dữ liệu thu được sau khi dữ liệu sai phân được mã hóa và ký, thông tin tiêu đề bao gồm thông tin tiêu đề mã hóa dữ liệu sai phân hoặc thông tin tiêu đề bao gồm thông tin tiêu đề mã hóa dữ liệu sai phân và thông tin tiêu đề mã hóa dữ liệu đích, thông tin tiêu đề mã hóa dữ liệu sai phân được sử dụng bởi thiết bị đầu cuối di động để giải mã dữ liệu được mã hóa sai phân để thu được dữ liệu sai phân, và thông tin tiêu đề mã hóa dữ liệu đích được sử dụng bởi thiết bị đầu cuối di động để xác minh chữ ký của dữ liệu được mã hóa sai phân; và

gửi, bởi máy chủ thứ nhất, dữ liệu OTA tới thiết bị đầu cuối di động, trong đó dữ liệu OTA được sử dụng để nâng cấp phiên bản cũ của phần mềm hệ thống của thiết bị đầu cuối di động thành phiên bản mới.

2. Phương pháp theo điểm 1, trong đó bước tạo ra, bởi máy chủ thứ nhất, dữ liệu OTA bao gồm:

mã hóa, bởi máy chủ thứ nhất, dữ liệu sai phân để thu được dữ liệu được mã hóa

sai phân, hoặc mã hóa và ký dữ liệu sai phân để thu được dữ liệu được mã hóa sai phân;

tạo ra, bởi máy chủ thứ nhất, thông tin tiêu đề cho dữ liệu được mã hóa sai phân; và

tích hợp, bởi máy chủ thứ nhất, dữ liệu được mã hóa sai phân và thông tin tiêu đề để thu được dữ liệu OTA.

3. Phương pháp theo điểm 1 hoặc 2, trong đó trước khi thu được, bởi máy chủ thứ nhất, phiên bản mới của dữ liệu được mã hóa và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống được áp dụng cho thiết bị đầu cuối di động, phương pháp này còn bao gồm các bước:

nhận, bởi máy chủ thứ nhất, phiên bản cũ của thông tin mà thuộc về phần mềm hệ thống và được gửi bởi thiết bị đầu cuối di động; và

xác định, bởi máy chủ thứ nhất dựa trên phiên bản cũ của thông tin của phần mềm hệ thống, phiên bản cũ của dữ liệu được mã hóa từ một hoặc nhiều phiên bản lịch sử của dữ liệu được mã hóa của phần mềm hệ thống.

4. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó bước giải mã, bởi máy chủ thứ nhất, phiên bản mới của dữ liệu được mã hóa để thu được phiên bản mới của dữ liệu ban đầu, và giải mã phiên bản cũ của dữ liệu được mã hóa để thu được phiên bản cũ của dữ liệu ban đầu bao gồm:

loại bỏ, bởi máy chủ thứ nhất, dữ liệu chữ ký từ phiên bản mới của dữ liệu được mã hóa, để thu được phiên bản mới của dữ liệu ban đầu được mã hóa;

loại bỏ, bởi máy chủ thứ nhất, dữ liệu chữ ký từ phiên bản cũ của dữ liệu được mã hóa, để thu được phiên bản cũ của dữ liệu ban đầu được mã hóa; và

giải mã, bởi máy chủ thứ nhất, phiên bản mới của dữ liệu ban đầu được mã hóa để thu được phiên bản mới của dữ liệu ban đầu, và giải mã phiên bản cũ của dữ liệu ban đầu được mã hóa để thu được phiên bản cũ của dữ liệu ban đầu.

5. Phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 3, trong đó phiên bản cũ của dữ liệu ban đầu bao gồm phiên bản cũ của khối dữ liệu tệp, phiên bản cũ của khối dữ liệu hàm băm, và phiên bản cũ của khối dữ liệu kiểm tra; phiên bản mới của dữ liệu ban đầu bao gồm phiên bản mới của khối dữ liệu tệp, phiên bản mới của khối dữ liệu hàm băm, và phiên bản mới của khối dữ liệu kiểm tra; phiên bản cũ của khối dữ liệu

kiểm tra là dữ liệu chữ ký của giá trị hàm băm gốc trong phiên bản cũ của khối dữ liệu hàm băm, và phiên bản cũ của khối dữ liệu hàm băm được sử dụng để kiểm tra phiên bản cũ của khối dữ liệu tệp; và phiên bản mới của khối dữ liệu kiểm tra là dữ liệu chữ ký của giá trị hàm băm gốc trong phiên bản mới của khối dữ liệu hàm băm, và phiên bản mới của khối dữ liệu hàm băm được sử dụng để kiểm tra phiên bản mới của khối dữ liệu tệp; và

bước thực hiện, bởi máy chủ thứ nhất, phép sai phân trên phiên bản mới của dữ liệu ban đầu và phiên bản cũ của dữ liệu ban đầu, để thu được dữ liệu sai phân bao gồm:

loại bỏ, bởi máy chủ thứ nhất, phiên bản cũ của khối dữ liệu hàm băm khỏi phiên bản cũ của dữ liệu ban đầu;

loại bỏ, bởi máy chủ thứ nhất, phiên bản mới của khối dữ liệu hàm băm khỏi phiên bản mới của dữ liệu ban đầu; và

thực hiện, bởi máy chủ thứ nhất, phép sai phân trên khối dữ liệu dựa trên phiên bản cũ của dữ liệu ban đầu mà phiên bản cũ của khối dữ liệu hàm băm được loại bỏ khỏi đó và phiên bản mới của dữ liệu ban đầu mà phiên bản mới của khối dữ liệu hàm băm được loại bỏ khỏi đó, để thu được dữ liệu sai phân.

6. Phương pháp nâng cấp qua vô tuyến, bao gồm các bước:

nhận, bởi thiết bị đầu cuối di động, dữ liệu OTA được gửi bởi máy chủ thứ nhất, trong đó dữ liệu OTA được sử dụng để nâng cấp phiên bản cũ của phần mềm hệ thống trên thiết bị đầu cuối di động thành phiên bản mới, dữ liệu OTA bao gồm thông tin tiêu đề và dữ liệu được mã hóa sai phân, thông tin tiêu đề bao gồm thông tin tiêu đề mã hóa dữ liệu sai phân hoặc thông tin tiêu đề bao gồm thông tin tiêu đề mã hóa dữ liệu sai phân và thông tin tiêu đề mã hóa dữ liệu đích, thông tin tiêu đề mã hóa dữ liệu sai phân được sử dụng bởi thiết bị đầu cuối di động để thu được dữ liệu sai phân thông qua hoạt động giải mã, thông tin tiêu đề mã hóa dữ liệu đích được sử dụng bởi thiết bị đầu cuối di động để xác minh chữ ký của dữ liệu được mã hóa sai phân, và dữ liệu được mã hóa sai phân là dữ liệu thu được sau khi dữ liệu sai phân được mã hóa hoặc dữ liệu được mã hóa sai phân là dữ liệu thu được sau khi dữ liệu sai phân được mã hóa và ký;

giải mã, bởi thiết bị đầu cuối di động, dữ liệu được mã hóa sai phân dựa trên

thông tin tiêu đề mã hóa dữ liệu sai phân, để thu được dữ liệu sai phân;

thu được, bởi thiết bị đầu cuối di động, phiên bản cũ của dữ liệu ban đầu của phần mềm hệ thống;

tích hợp, bởi thiết bị đầu cuối di động, phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu; và

nâng cấp, bởi thiết bị đầu cuối di động, phiên bản cũ của phần mềm hệ thống thành phiên bản mới bằng cách sử dụng phiên bản mới của dữ liệu ban đầu.

7. Phương pháp theo điểm 6, trong đó trước khi nhận, bởi thiết bị đầu cuối di động, dữ liệu OTA được gửi bởi máy chủ thứ nhất, phương pháp này còn bao gồm bước:

gửi, bởi thiết bị đầu cuối di động, phiên bản cũ của thông tin của phần mềm hệ thống tới máy chủ thứ nhất, trong đó phiên bản cũ của thông tin của phần mềm hệ thống được sử dụng bởi máy chủ thứ nhất để xác định phiên bản cũ của dữ liệu được mã hóa từ một hoặc nhiều phiên bản lịch sử của dữ liệu được mã hóa.

8. Phương pháp theo điểm 6 hoặc 7, trong đó thiết bị đầu cuối di động bao gồm môi trường thực thi tin cậy, TEE, và môi trường thực thi phong phú, REE, ứng dụng tin cậy, TA, chạy trong TEE, và ứng dụng khách, CA, thứ nhất và ứng dụng cập nhật chạy trong REE.

9. Phương pháp theo điểm 8, trong đó bước giải mã, bởi thiết bị đầu cuối di động, dữ liệu được mã hóa sai phân dựa trên thông tin tiêu đề mã hóa dữ liệu sai phân, để thu được dữ liệu sai phân bao gồm:

trích xuất, bởi ứng dụng cập nhật, dữ liệu được mã hóa sai phân và thông tin tiêu đề mã hóa dữ liệu sai phân từ dữ liệu OTA, trong đó thông tin tiêu đề mã hóa dữ liệu sai phân bao gồm thuật toán mã hóa của dữ liệu được mã hóa sai phân và giá trị ngẫu nhiên mã hóa được sử dụng trong hoạt động mã hóa;

chuyển tiếp, bởi ứng dụng cập nhật, dữ liệu được mã hóa sai phân, thuật toán tiêu đề mã hóa của dữ liệu được mã hóa sai phân, và giá trị ngẫu nhiên mã hóa được sử dụng trong hoạt động mã hóa tới TA bằng cách sử dụng CA thứ nhất; và

giải mã, bởi TA, dữ liệu được mã hóa sai phân dựa trên thông tin tiêu đề mã hóa dữ liệu sai phân, để thu được dữ liệu sai phân.

10. Phương pháp theo điểm 9, trong đó khi thông tin tiêu đề bao gồm thông tin tiêu đề mã hóa dữ liệu sai phân và thông tin tiêu đề mã hóa dữ liệu đích, trước khi tích hợp,

bởi thiết bị đầu cuối di động, phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu, phương pháp này còn bao gồm các bước:

trích xuất, bởi ứng dụng cập nhật, thông tin tiêu đề mã hóa dữ liệu đích từ thông tin tiêu đề, trong đó thông tin tiêu đề mã hóa dữ liệu đích bao gồm thông tin thuật toán tóm lược thông điệp của dữ liệu được mã hóa sai phân và thông tin thuật toán mã hóa dùng cho chữ ký;

chuyển tiếp, bởi ứng dụng cập nhật, thông tin thuật toán tóm lược thông điệp của dữ liệu được mã hóa sai phân và thông tin thuật toán mã hóa dùng cho chữ ký tới TA bằng cách sử dụng CA thứ nhất; và

xác minh, bởi TA, chữ ký của dữ liệu được mã hóa sai phân dựa trên thông tin chữ ký, trong đó nếu việc xác minh thành công, thiết bị đầu cuối di động tích hợp phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu.

11. Phương pháp theo điểm 8, trong đó bước thu được, bởi thiết bị đầu cuối di động, phiên bản cũ của dữ liệu ban đầu của phần mềm hệ thống bao gồm:

chuyển tiếp, bởi ứng dụng cập nhật, phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống tới TA bằng cách sử dụng CA thứ nhất; và

giải mã, bởi TA, phiên bản hiện tại của dữ liệu được mã hóa của phần mềm hệ thống để thu được phiên bản cũ của dữ liệu ban đầu.

12. Phương pháp theo điểm bất kỳ trong số các điểm từ 9 đến 11, trong đó bước tích hợp, bởi thiết bị đầu cuối di động, phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu bao gồm:

tích hợp, bởi TA thứ nhất, phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu.

13. Phương pháp theo điểm 8, trong đó phiên bản cũ của dữ liệu ban đầu bao gồm phiên bản cũ của khối dữ liệu tệp, phiên bản cũ của khối dữ liệu hàm băm, và phiên bản cũ của khối dữ liệu kiểm tra, phiên bản cũ của khối dữ liệu kiểm tra là dữ liệu chữ ký của giá trị hàm băm gốc trong phiên bản cũ của khối dữ liệu hàm băm, và phiên bản cũ của khối dữ liệu hàm băm được sử dụng để kiểm tra phiên bản cũ của khối dữ liệu tệp; và

bước tích hợp, bởi TA, phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai

phân và phiên bản cũ của dữ liệu ban đầu bao gồm:

tích hợp, bởi TA, phiên bản mới của khối dữ liệu tệp và phiên bản mới của khối dữ liệu kiểm tra dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu; và

tính toán, bởi TA, cây hàm băm của phiên bản mới của khối dữ liệu tệp, và tạo ra phiên bản mới của khối dữ liệu hàm băm, để thu được phiên bản mới của dữ liệu ban đầu, trong đó phiên bản mới của dữ liệu ban đầu bao gồm phiên bản mới của khối dữ liệu tệp, phiên bản mới của khối dữ liệu hàm băm, và phiên bản mới của khối dữ liệu kiểm tra, phiên bản mới của khối dữ liệu kiểm tra là dữ liệu chữ ký của giá trị hàm băm gốc trong phiên bản mới của khối dữ liệu hàm băm, và phiên bản mới của khối dữ liệu hàm băm được sử dụng để kiểm tra phiên bản mới của khối dữ liệu tệp.

14. Máy chủ bao gồm bộ nhớ, bộ thu phát, và ít nhất một bộ xử lý, trong đó bộ nhớ lưu trữ mã chương trình, bộ nhớ, bộ thu phát, và ít nhất một bộ xử lý truyền thông với nhau, và bộ xử lý chạy mã để ra lệnh cho máy chủ thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 1 đến 5.

15. Thiết bị đầu cuối di động bao gồm bộ nhớ, bộ thu phát, và ít nhất một bộ xử lý, trong đó bộ nhớ lưu trữ mã chương trình, bộ nhớ, bộ thu phát, và ít nhất một bộ xử lý truyền thông với nhau, và bộ xử lý chạy mã để ra lệnh cho thiết bị đầu cuối di động thực hiện phương pháp theo điểm bất kỳ trong số các điểm từ 6 đến 13.

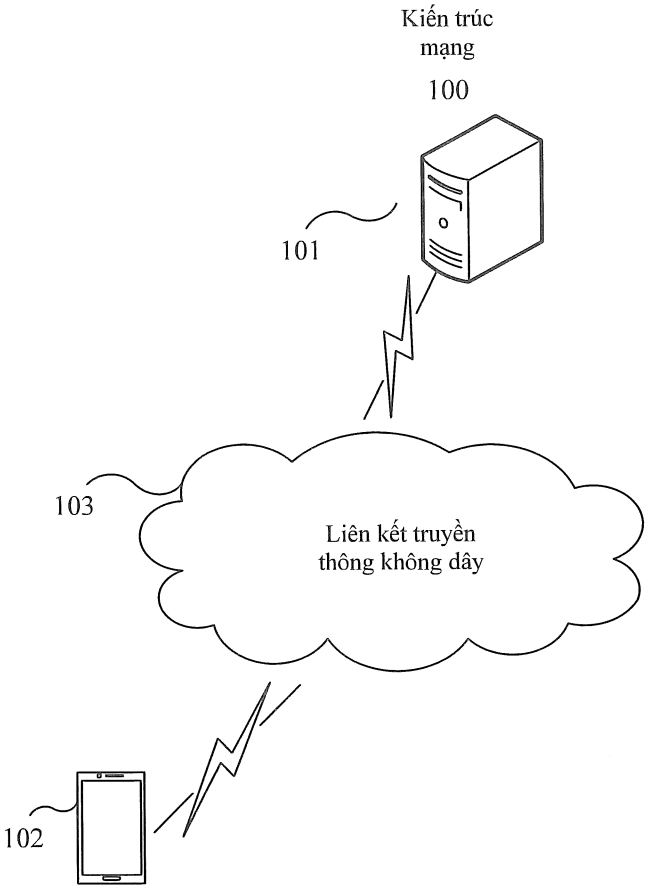

Kiến trúc
mạng
100
101
103
Liên kết truyền
thông không dây
102

FIG. 1

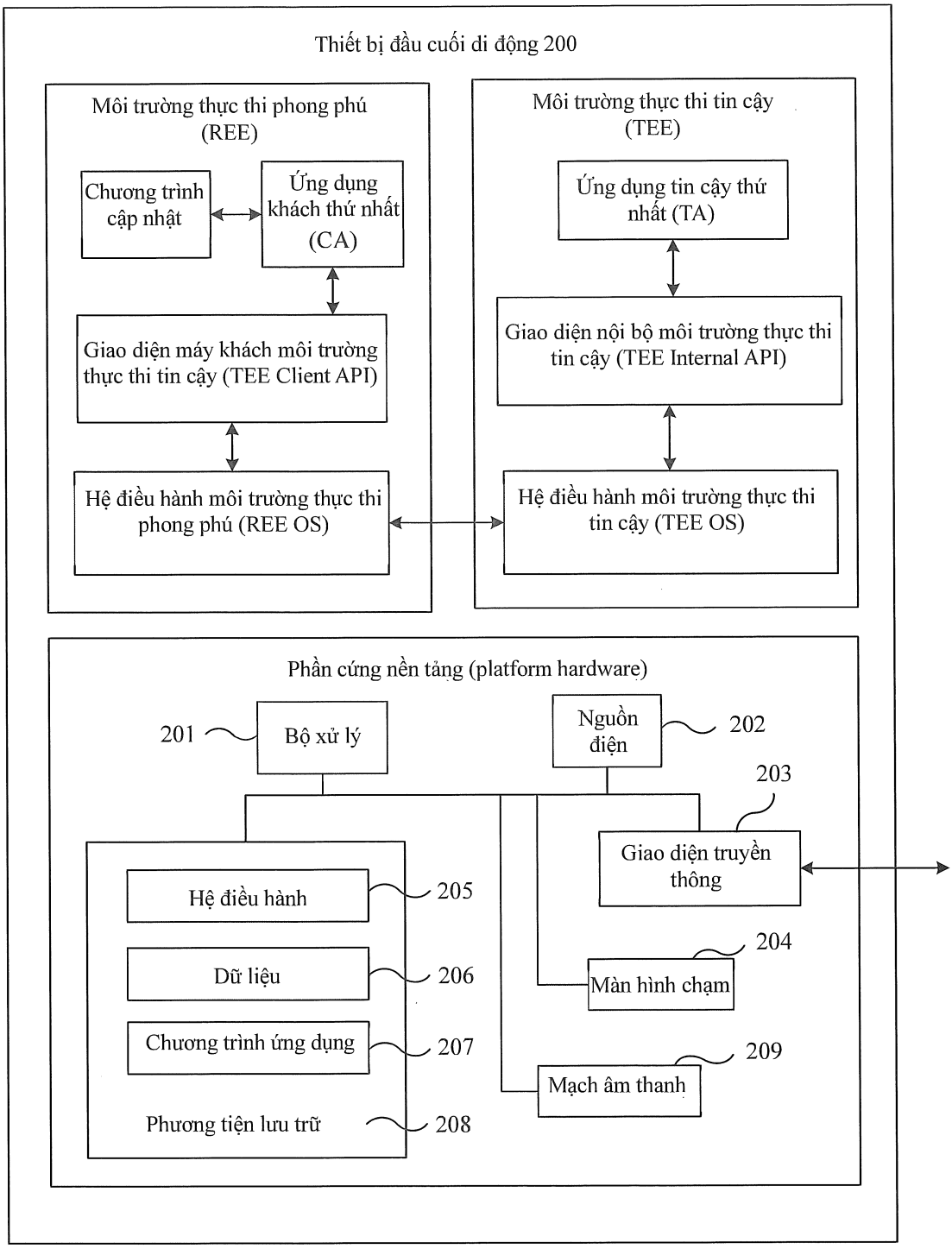

Thiết bị đầu cuối di động 200
Môi trường thực thi phong phú (REE)
Chương trình cập nhật
Ứng dụng khách thứ nhất (CA)
Giao diện máy khách môi trường thực thi tin cậy (TEE Client API)
Hệ điều hành môi trường thực thi phong phú (REE OS)
Môi trường thực thi tin cậy (TEE)
Ứng dụng tin cậy thứ nhất (TA)
Giao diện nội bộ môi trường thực thi tin cậy (TEE Internal API)
Hệ điều hành môi trường thực thi tin cậy (TEE OS)
Phần cứng nền tảng (platform hardware)
201
Bộ xử lý
Nguồn điện
202
203
Giao diện truyền thông
Hệ điều hành
205
Dữ liệu
206
Chương trình ứng dụng
207
Phương tiện lưu trữ
208
204
Màn hình chạm
209
Mạch âm thanh

FIG. 2

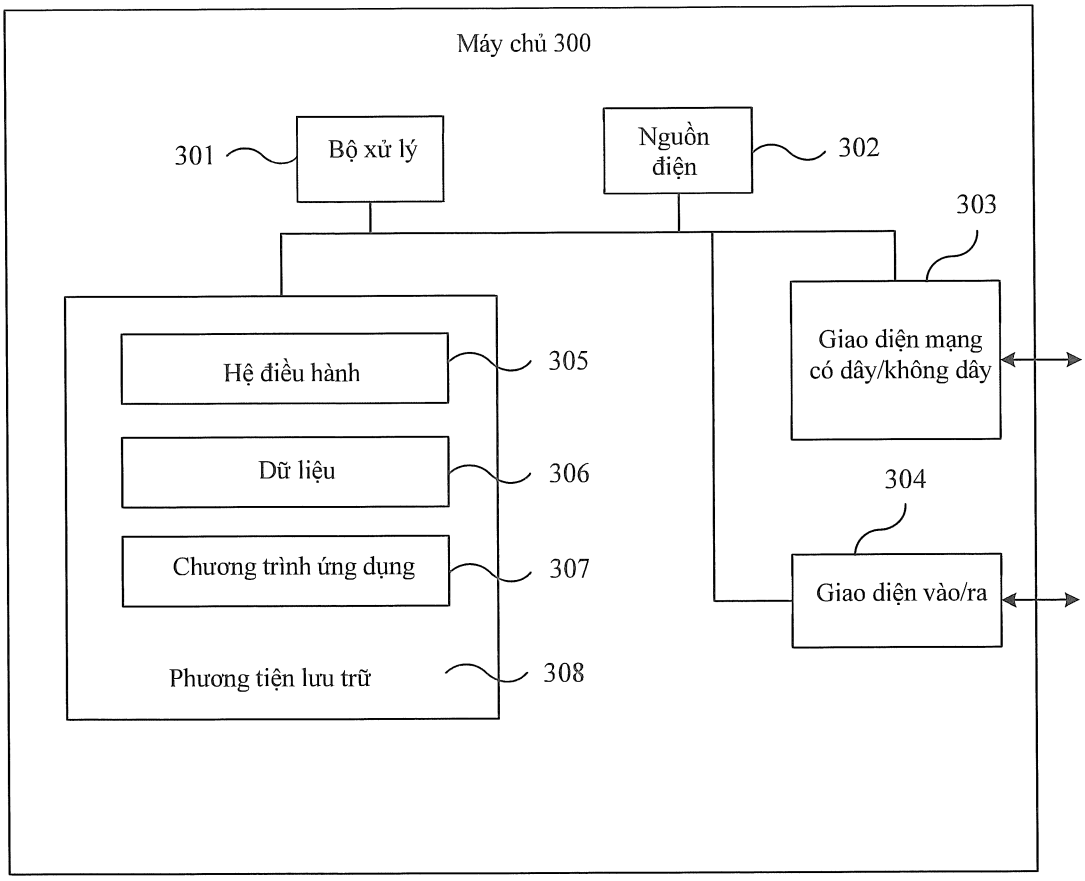

Máy chủ 300
301
Bộ xử lý
Nguồn điện
302
303
Giao diện mạng có dây/không dây
Hệ điều hành
305
Dữ liệu
306
304
Chương trình ứng dụng
307
Giao diện vào/ra
Phương tiện lưu trữ
308

FIG. 3

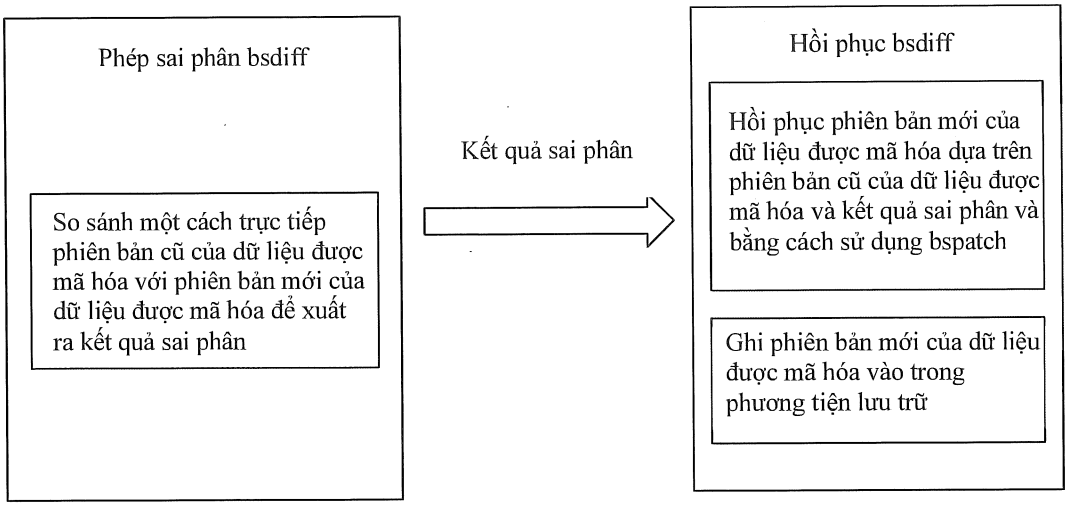

Phép sai phân bsdiff
So sánh một cách trực tiếp phiên bản cũ của dữ liệu được mã hóa với phiên bản mới của dữ liệu được mã hóa để xuất ra kết quả sai phân
Kết quả sai phân
Hồi phục bsdiff
Hồi phục phiên bản mới của dữ liệu được mã hóa dựa trên phiên bản cũ của dữ liệu được mã hóa và kết quả sai phân và bằng cách sử dụng bspatch
Ghi phiên bản mới của dữ liệu được mã hóa vào trong phương tiện lưu trữ

FIG. 4a

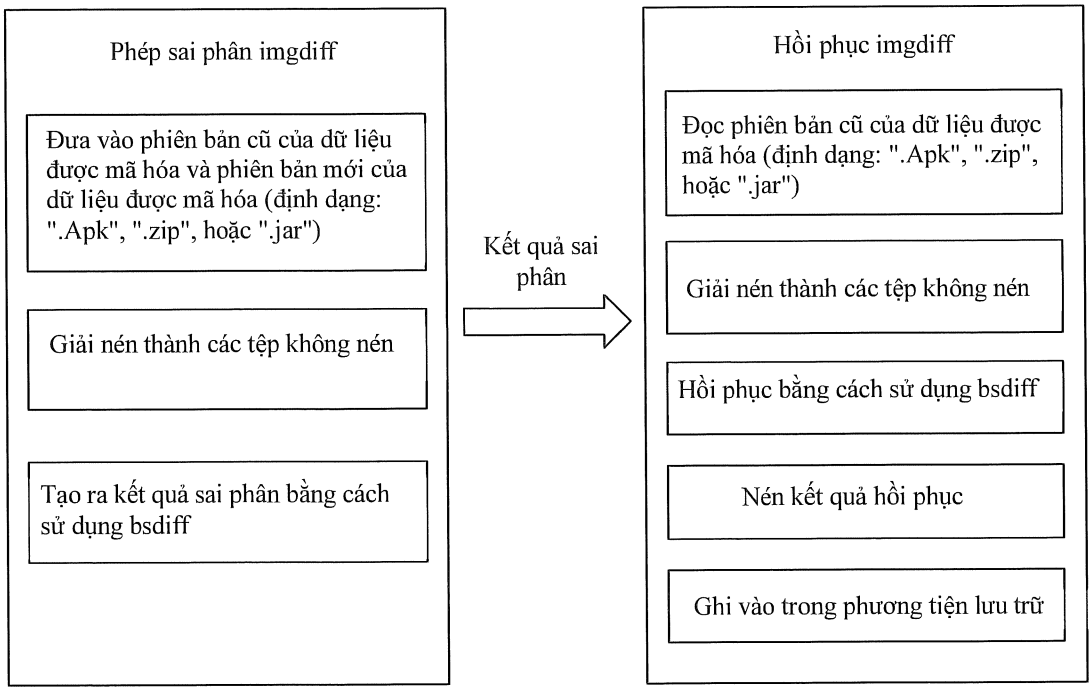

Phép sai phân imgdiff
Đưa vào phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa (định dạng: ".Apk", ".zip", hoặc ".jar")
Giải nén thành các tệp không nén
Tạo ra kết quả sai phân bằng cách sử dụng bsdiff
Kết quả sai phân
Hồi phục imgdiff
Đọc phiên bản cũ của dữ liệu được mã hóa (định dạng: ".Apk", ".zip", hoặc ".jar")
Giải nén thành các tệp không nén
Hồi phục bằng cách sử dụng bsdiff
Nén kết quả hồi phục
Ghi vào trong phương tiện lưu trữ

FIG. 4b

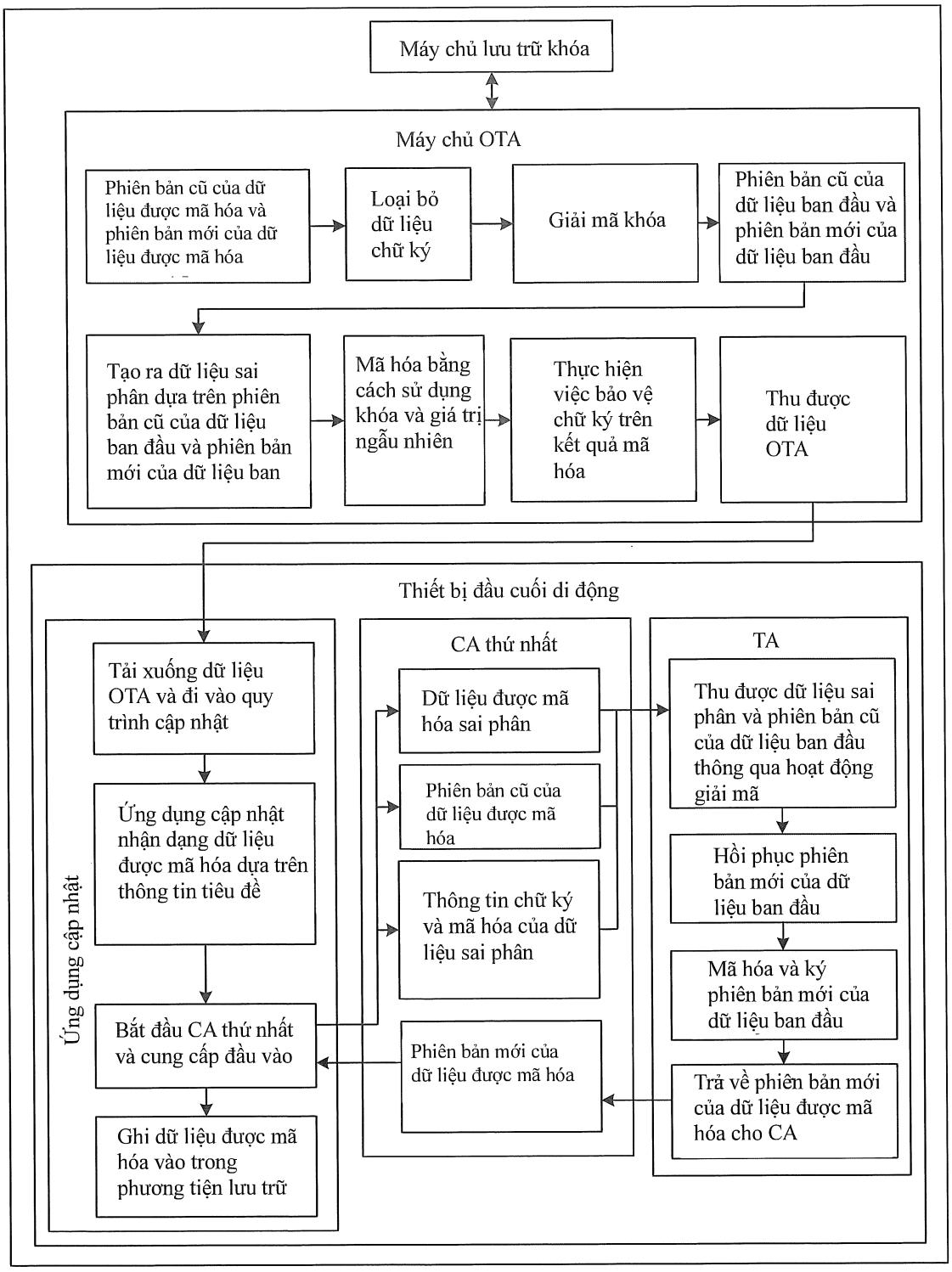

Máy chủ lưu trữ khóa
Máy chủ OTA
Phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa
Loại bỏ dữ liệu chữ ký
Giải mã khóa
Phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu
Tạo ra dữ liệu sai phân dựa trên phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban
Mã hóa bằng cách sử dụng khóa và giá trị ngẫu nhiên
Thực hiện việc bảo vệ chữ ký trên kết quả mã hóa
Thu được dữ liệu OTA
Thiết bị đầu cuối di động
Ứng dụng cập nhật
Tải xuống dữ liệu OTA và đi vào quy trình cập nhật
Ứng dụng cập nhật nhận dạng dữ liệu được mã hóa dựa trên thông tin tiêu đề
Bắt đầu CA thứ nhất và cung cấp đầu vào
Ghi dữ liệu được mã hóa vào trong phương tiện lưu trữ
CA thứ nhất
Dữ liệu được mã hóa sai phân
Phiên bản cũ của dữ liệu được mã hóa
Thông tin chữ ký và mã hóa của dữ liệu sai phân
Phiên bản mới của dữ liệu được mã hóa
TA
Thu được dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu thông qua hoạt động giải mã
Hồi phục phiên bản mới của dữ liệu ban đầu
Mã hóa và ký phiên bản mới của dữ liệu ban đầu
Trả về phiên bản mới của dữ liệu được mã hóa cho CA

FIG. 5a

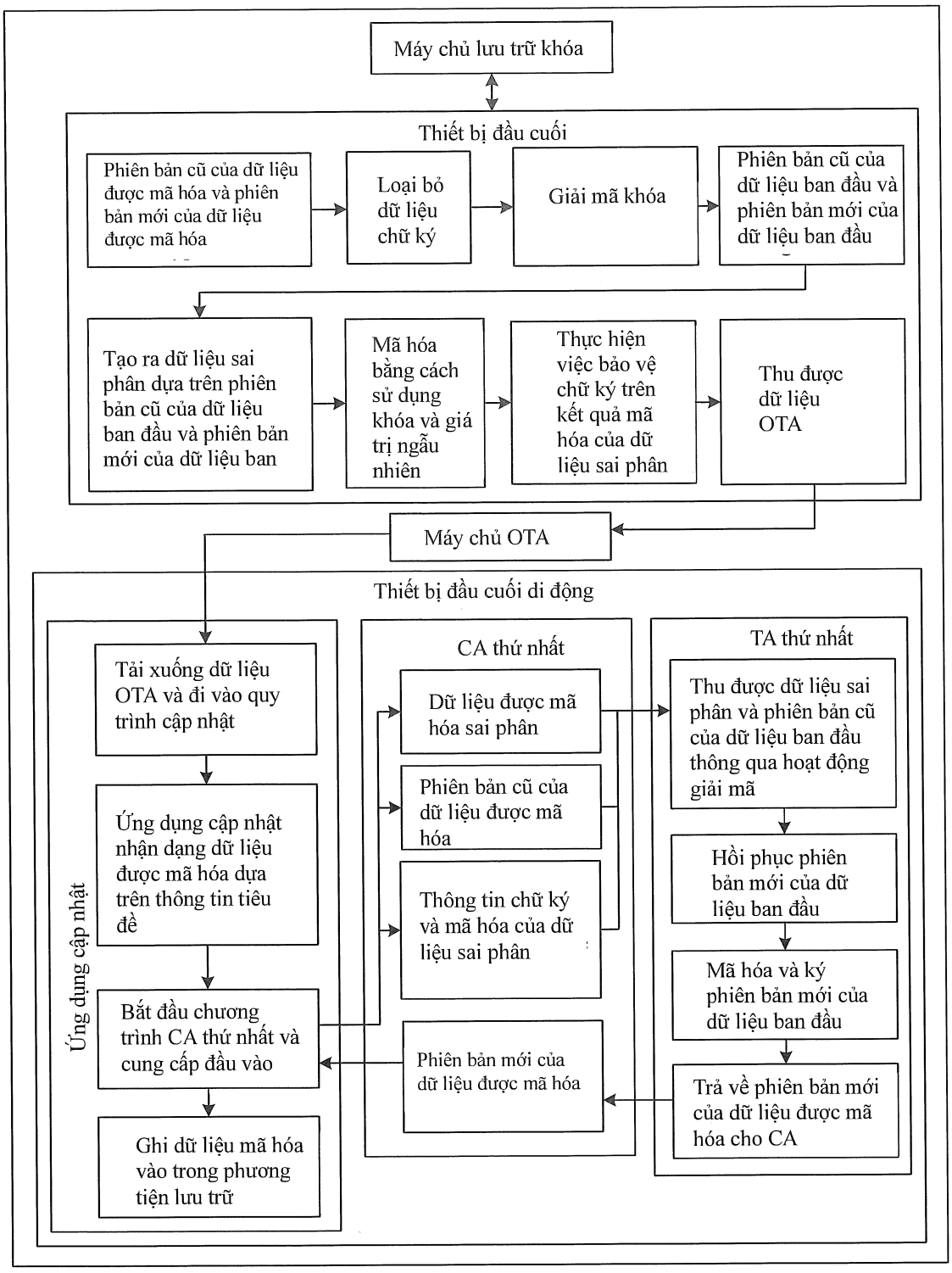

Máy chủ lưu trữ khóa
Thiết bị đầu cuối
Phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa
Loại bỏ dữ liệu chữ ký
Giải mã khóa
Phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu
Tạo ra dữ liệu sai phân dựa trên phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban
Mã hóa bằng cách sử dụng khóa và giá trị ngẫu nhiên
Thực hiện việc bảo vệ chữ ký trên kết quả mã hóa của dữ liệu sai phân
Thu được dữ liệu OTA
Máy chủ OTA
Thiết bị đầu cuối di động
Ứng dụng cập nhật
Tải xuống dữ liệu OTA và đi vào quy trình cập nhật
Ứng dụng cập nhật nhận dạng dữ liệu được mã hóa dựa trên thông tin tiêu đề
Bắt đầu chương trình CA thứ nhất và cung cấp đầu vào
Ghi dữ liệu mã hóa vào trong phương tiện lưu trữ
CA thứ nhất
Dữ liệu được mã hóa sai phân
Phiên bản cũ của dữ liệu được mã hóa
Thông tin chữ ký và mã hóa của dữ liệu sai phân
Phiên bản mới của dữ liệu được mã hóa
TA thứ nhất
Thu được dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu thông qua hoạt động giải mã
Hồi phục phiên bản mới của dữ liệu ban đầu
Mã hóa và ký phiên bản mới của dữ liệu ban đầu
Trả về phiên bản mới của dữ liệu được mã hóa cho CA

FIG. 5b

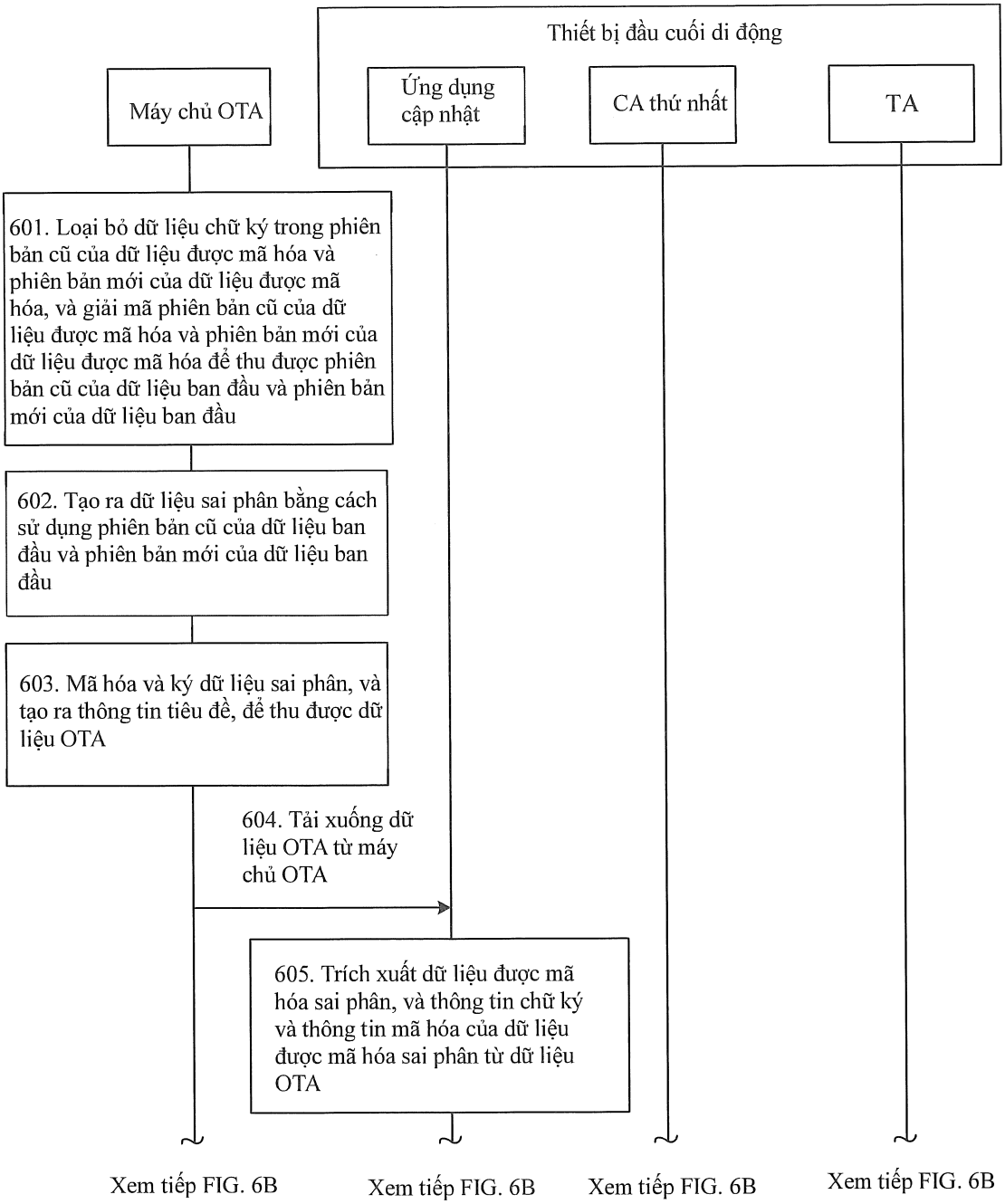

Thiết bị đầu cuối di động
Máy chủ OTA
Ứng dụng cập nhật
CA thứ nhất
TA
601. Loại bỏ dữ liệu chữ ký trong phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa, và giải mã phiên bản cũ của dữ liệu được mã hóa và phiên bản mới của dữ liệu được mã hóa để thu được phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu
602. Tạo ra dữ liệu sai phân bằng cách sử dụng phiên bản cũ của dữ liệu ban đầu và phiên bản mới của dữ liệu ban đầu
603. Mã hóa và ký dữ liệu sai phân, và tạo ra thông tin tiêu đề, để thu được dữ liệu OTA
604. Tải xuống dữ liệu OTA từ máy chủ OTA
605. Trích xuất dữ liệu được mã hóa sai phân, và thông tin chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân từ dữ liệu OTA
Xem tiếp FIG. 6B
Xem tiếp FIG. 6B
Xem tiếp FIG. 6B
Xem tiếp FIG. 6B

FIG. 6a

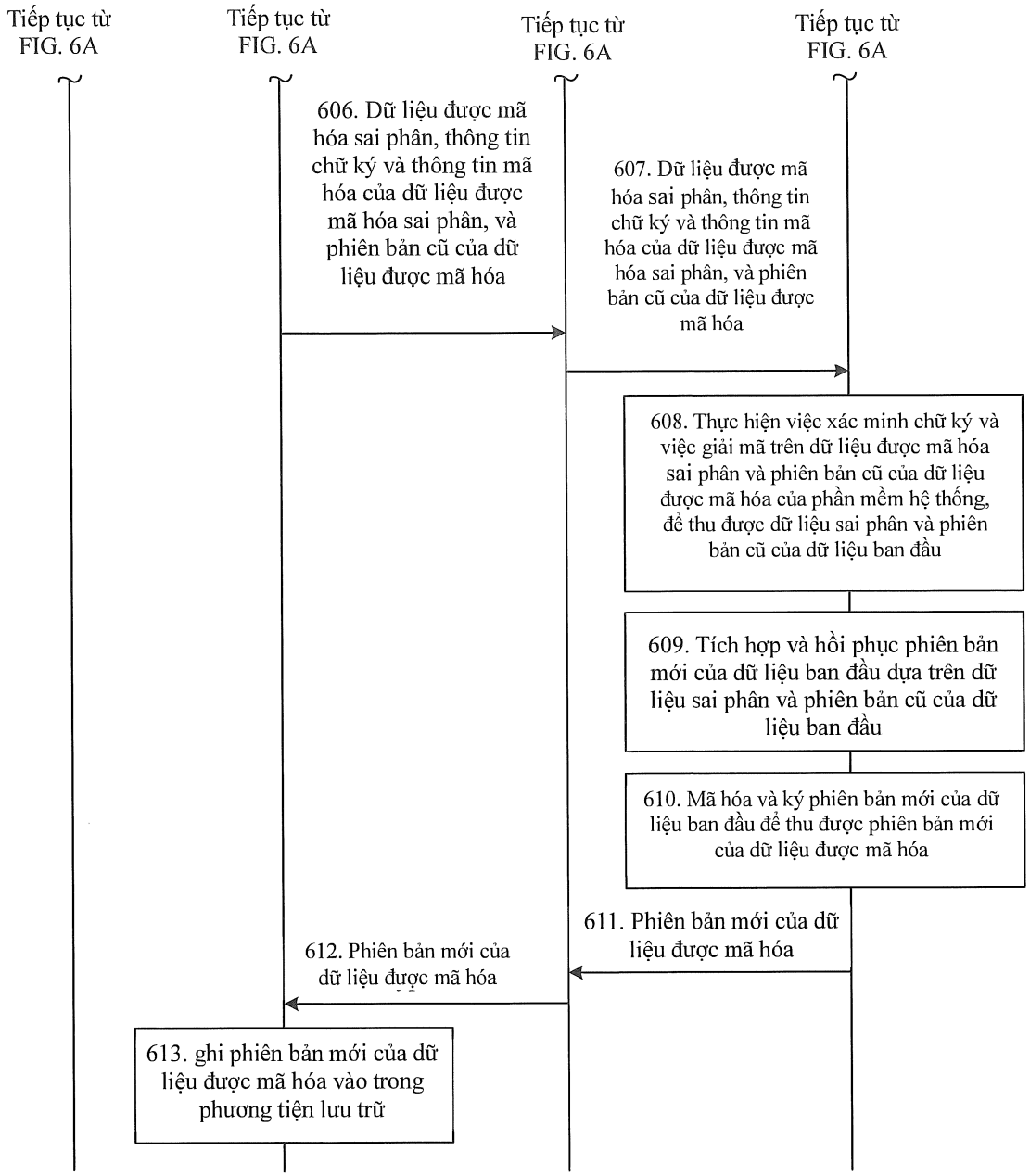

Tiếp tục từ FIG. 6A
Tiếp tục từ FIG. 6A
Tiếp tục từ FIG. 6A
Tiếp tục từ FIG. 6A
606. Dữ liệu được mã hóa sai phân, thông tin chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân, và phiên bản cũ của dữ liệu được mã hóa
607. Dữ liệu được mã hóa sai phân, thông tin chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân, và phiên bản cũ của dữ liệu được mã hóa
608. Thực hiện việc xác minh chữ ký và việc giải mã trên dữ liệu được mã hóa sai phân và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống, để thu được dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu
609. Tích hợp và hồi phục phiên bản mới của dữ liệu ban đầu dựa trên dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu
610. Mã hóa và ký phiên bản mới của dữ liệu ban đầu để thu được phiên bản mới của dữ liệu được mã hóa
611. Phiên bản mới của dữ liệu được mã hóa
612. Phiên bản mới của dữ liệu được mã hóa
613. ghi phiên bản mới của dữ liệu được mã hóa vào trong phương tiện lưu trữ

FIG. 6b

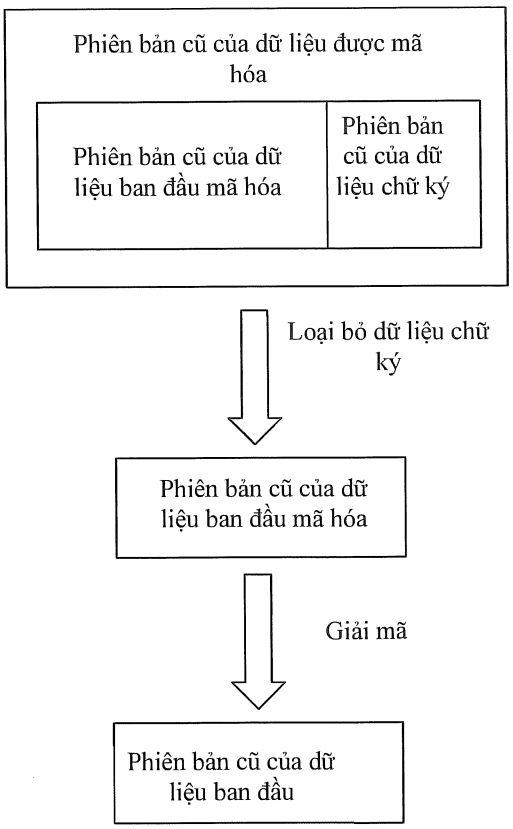

Phiên bản cũ của dữ liệu được mã hóa
Phiên bản cũ của dữ liệu ban đầu mã hóa
Phiên bản cũ của dữ liệu chữ ký
Loại bỏ dữ liệu chữ ký
Phiên bản cũ của dữ liệu ban đầu mã hóa
Giải mã
Phiên bản cũ của dữ liệu ban đầu

FIG. 7a

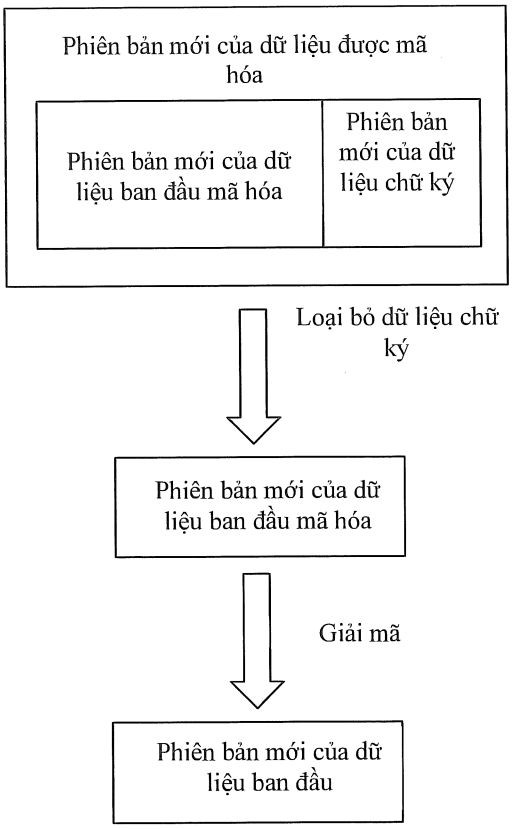

Phiên bản mới của dữ liệu được mã hóa
Phiên bản mới của dữ liệu ban đầu mã hóa
Phiên bản mới của dữ liệu chữ ký
Loại bỏ dữ liệu chữ ký
Phiên bản mới của dữ liệu ban đầu mã hóa
Giải mã
Phiên bản mới của dữ liệu ban đầu

FIG. 7b

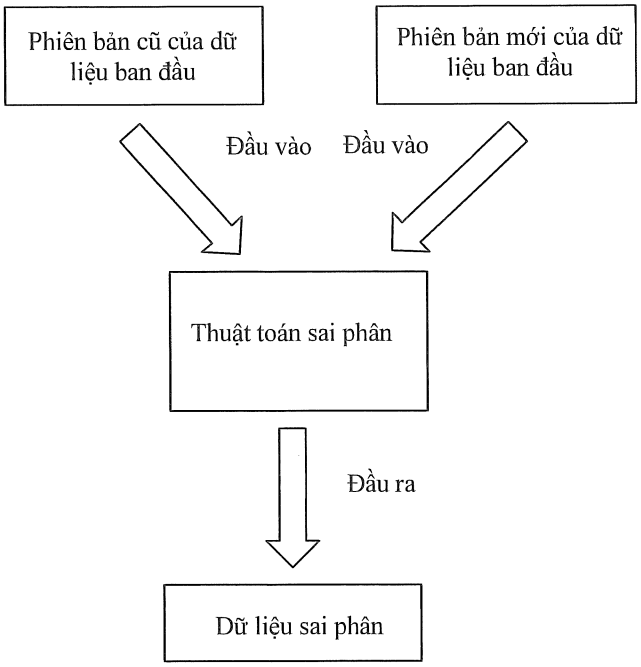

Phiên bản cũ của dữ liệu ban đầu
Phiên bản mới của dữ liệu ban đầu
Đầu vào
Đầu vào
Thuật toán sai phân
Đầu ra
Dữ liệu sai phân

FIG. 8

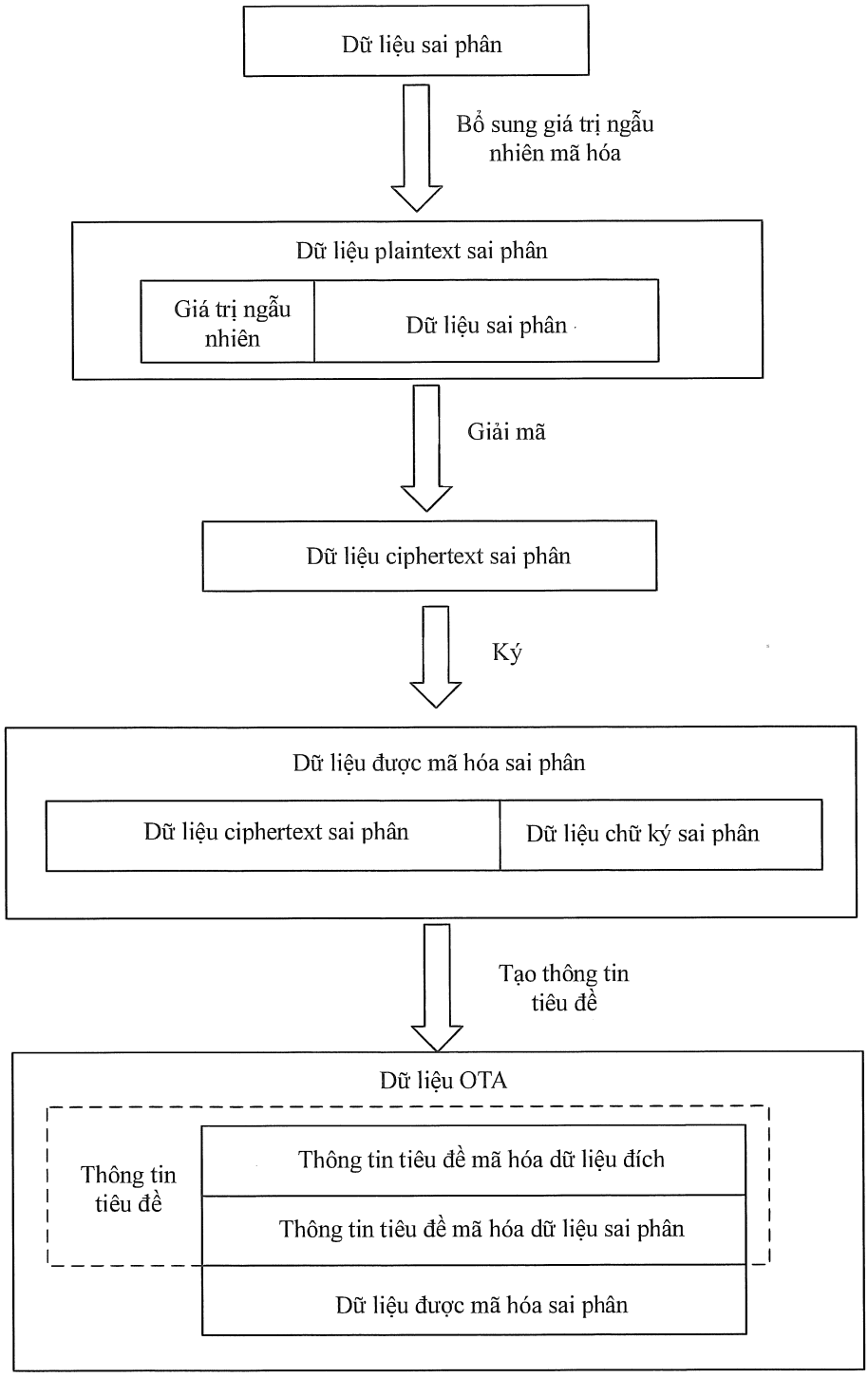

Dữ liệu sai phân
Bổ sung giá trị ngẫu nhiên mã hóa
Dữ liệu plaintext sai phân
Giá trị ngẫu nhiên
Dữ liệu sai phân
Giải mã
Dữ liệu ciphertext sai phân
Ký
Dữ liệu được mã hóa sai phân
Dữ liệu ciphertext sai phân
Dữ liệu chữ ký sai phân
Tạo thông tin tiêu đề
Dữ liệu OTA
Thông tin tiêu đề
Thông tin tiêu đề mã hóa dữ liệu đích
Thông tin tiêu đề mã hóa dữ liệu sai phân
Dữ liệu được mã hóa sai phân

FIG. 9

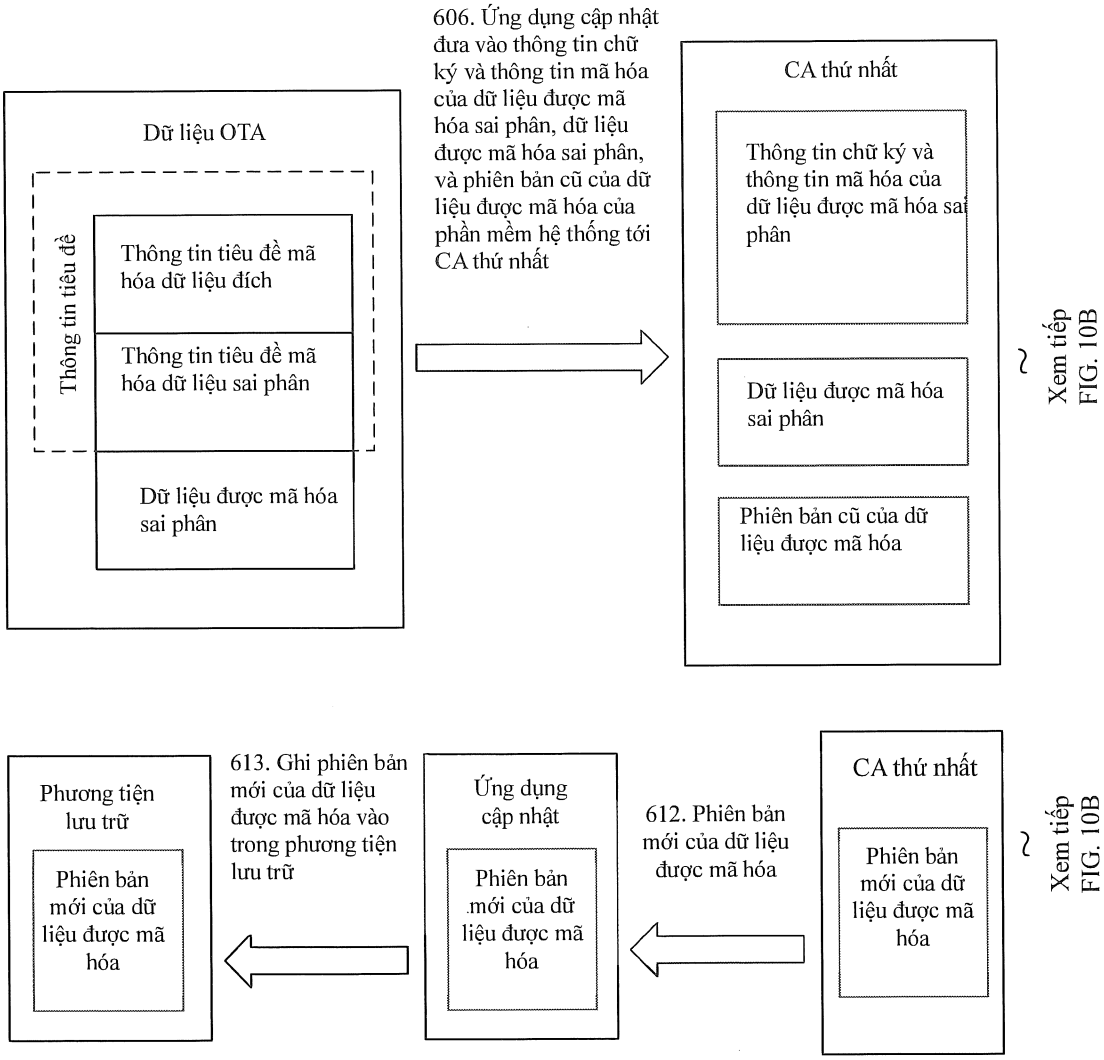

606. Ứng dụng cập nhật đưa vào thông tin chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân, dữ liệu được mã hóa sai phân, và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống tới CA thứ nhất
Dữ liệu OTA
Thông tin tiêu đề
Thông tin tiêu đề mã hóa dữ liệu đích
Thông tin tiêu đề mã hóa dữ liệu sai phân
Dữ liệu được mã hóa sai phân
CA thứ nhất
Thông tin chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân
Dữ liệu được mã hóa sai phân
Phiên bản cũ của dữ liệu được mã hóa
Xem tiếp FIG. 10B
Phương tiện lưu trữ
Phiên bản mới của dữ liệu được mã hóa
613. Ghi phiên bản mới của dữ liệu được mã hóa vào trong phương tiện lưu trữ
Ứng dụng cập nhật
Phiên bản mới của dữ liệu được mã hóa
612. Phiên bản mới của dữ liệu được mã hóa
CA thứ nhất
Phiên bản mới của dữ liệu được mã hóa
Xem tiếp FIG. 10B

FIG. 10A

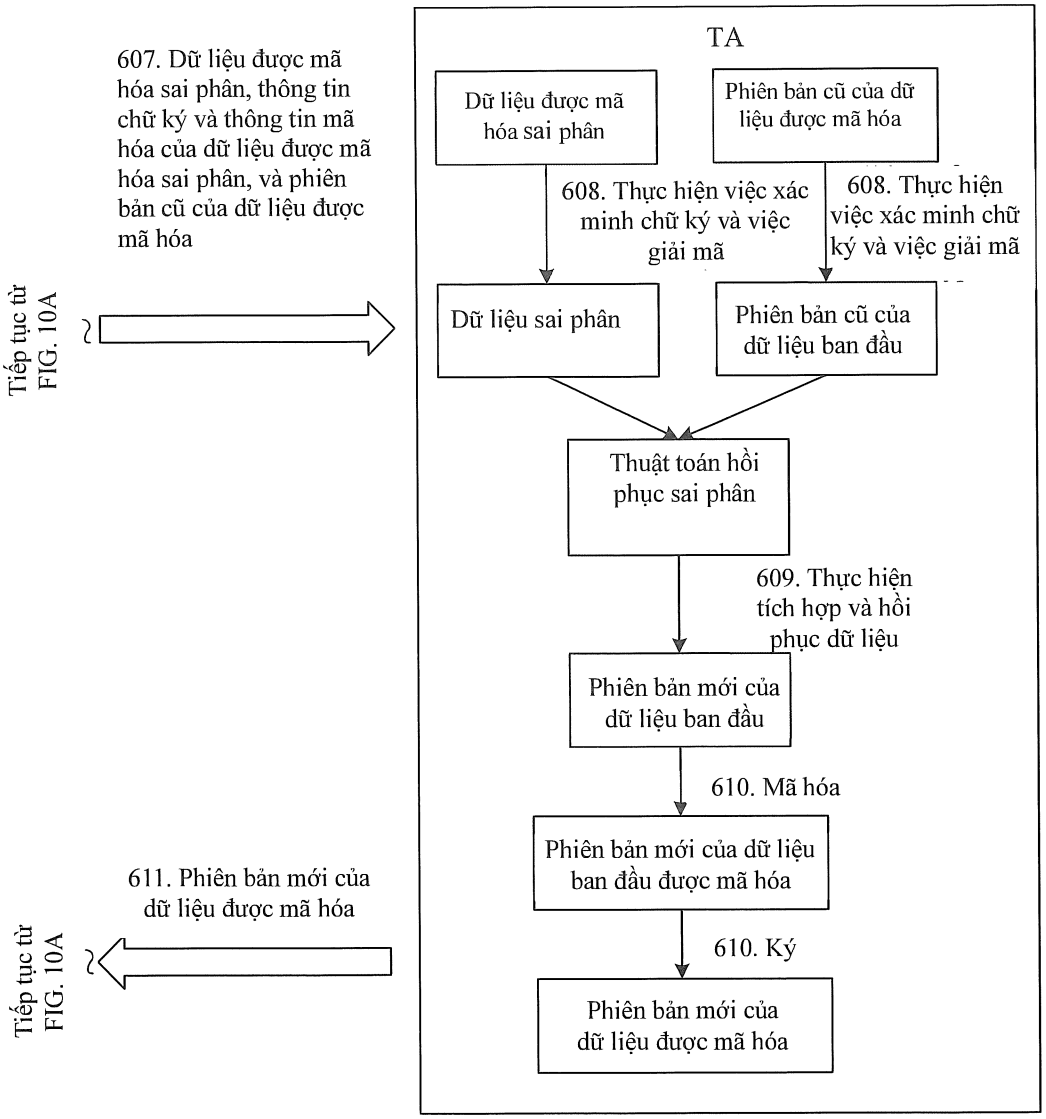

TA
607. Dữ liệu được mã hóa sai phân, thông tin chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân, và phiên bản cũ của dữ liệu được mã hóa
Tiếp tục từ FIG. 10A
Dữ liệu được mã hóa sai phân
Phiên bản cũ của dữ liệu được mã hóa
608. Thực hiện việc xác minh chữ ký và việc giải mã
608. Thực hiện việc xác minh chữ ký và việc giải mã
Dữ liệu sai phân
Phiên bản cũ của dữ liệu ban đầu
Thuật toán hồi phục sai phân
609. Thực hiện tích hợp và hồi phục dữ liệu
Phiên bản mới của dữ liệu ban đầu
610. Mã hóa
Phiên bản mới của dữ liệu ban đầu được mã hóa
610. Ký
Phiên bản mới của dữ liệu được mã hóa
611. Phiên bản mới của dữ liệu được mã hóa
Tiếp tục từ FIG. 10A

FIG. 10B

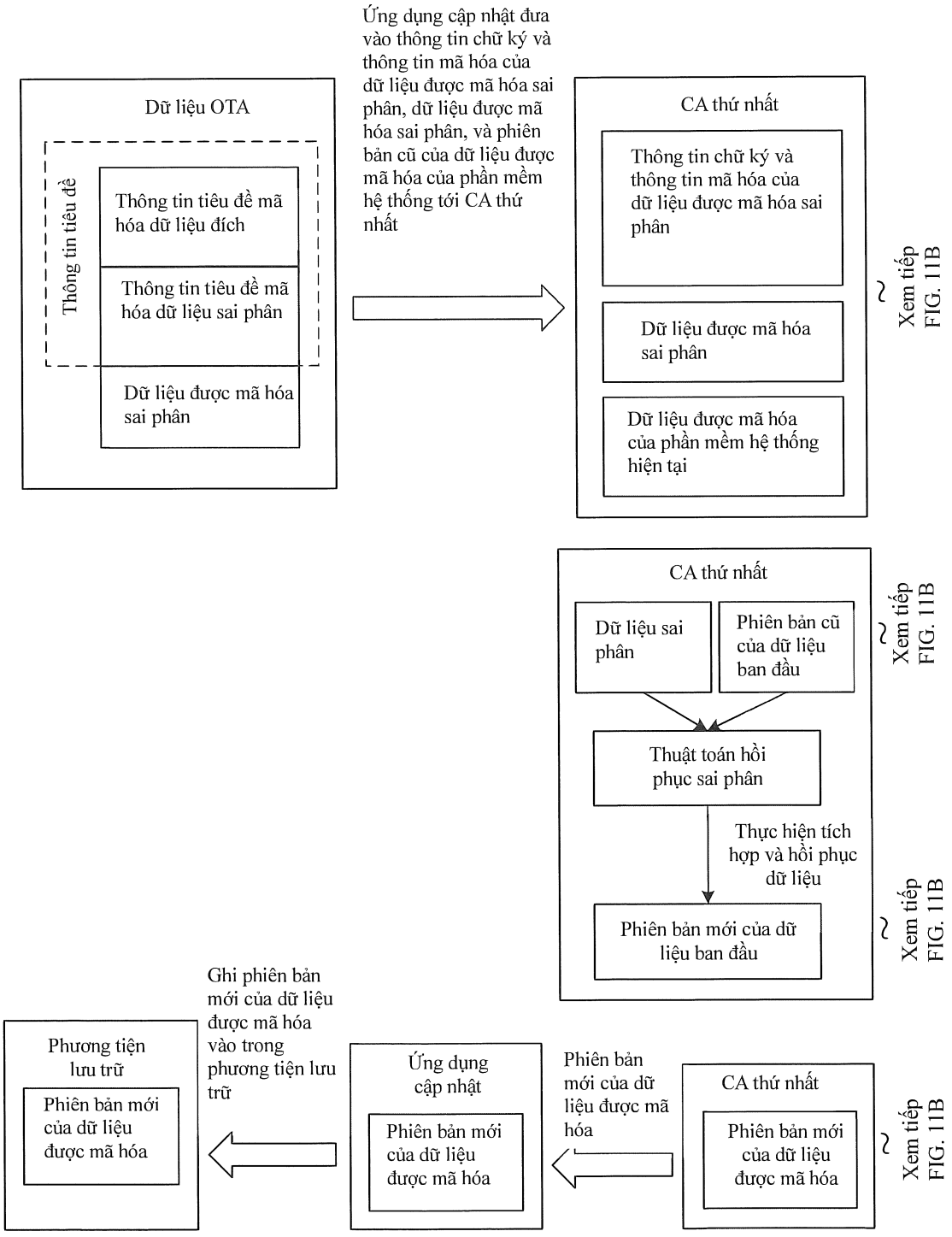

Dữ liệu OTA
Thông tin tiêu đề
Thông tin tiêu đề mã hóa dữ liệu đích
Thông tin tiêu đề mã hóa dữ liệu sai phân
Dữ liệu được mã hóa sai phân
Ứng dụng cập nhật đưa vào thông tin chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân, dữ liệu được mã hóa sai phân, và phiên bản cũ của dữ liệu được mã hóa của phần mềm hệ thống tới CA thứ nhất
CA thứ nhất
Thông tin chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân
Dữ liệu được mã hóa sai phân
Dữ liệu được mã hóa của phần mềm hệ thống hiện tại
Xem tiếp FIG. 11B
CA thứ nhất
Dữ liệu sai phân
Phiên bản cũ của dữ liệu ban đầu
Thuật toán hồi phục sai phân
Thực hiện tích hợp và hồi phục dữ liệu
Phiên bản mới của dữ liệu ban đầu
Xem tiếp FIG. 11B
Xem tiếp FIG. 11B
Ghi phiên bản mới của dữ liệu được mã hóa vào trong phương tiện lưu trữ
Phương tiện lưu trữ
Phiên bản mới của dữ liệu được mã hóa
Ứng dụng cập nhật
Phiên bản mới của dữ liệu được mã hóa
Phiên bản mới của dữ liệu được mã hóa
CA thứ nhất
Phiên bản mới của dữ liệu được mã hóa
Xem tiếp FIG. 11B

FIG. 11A

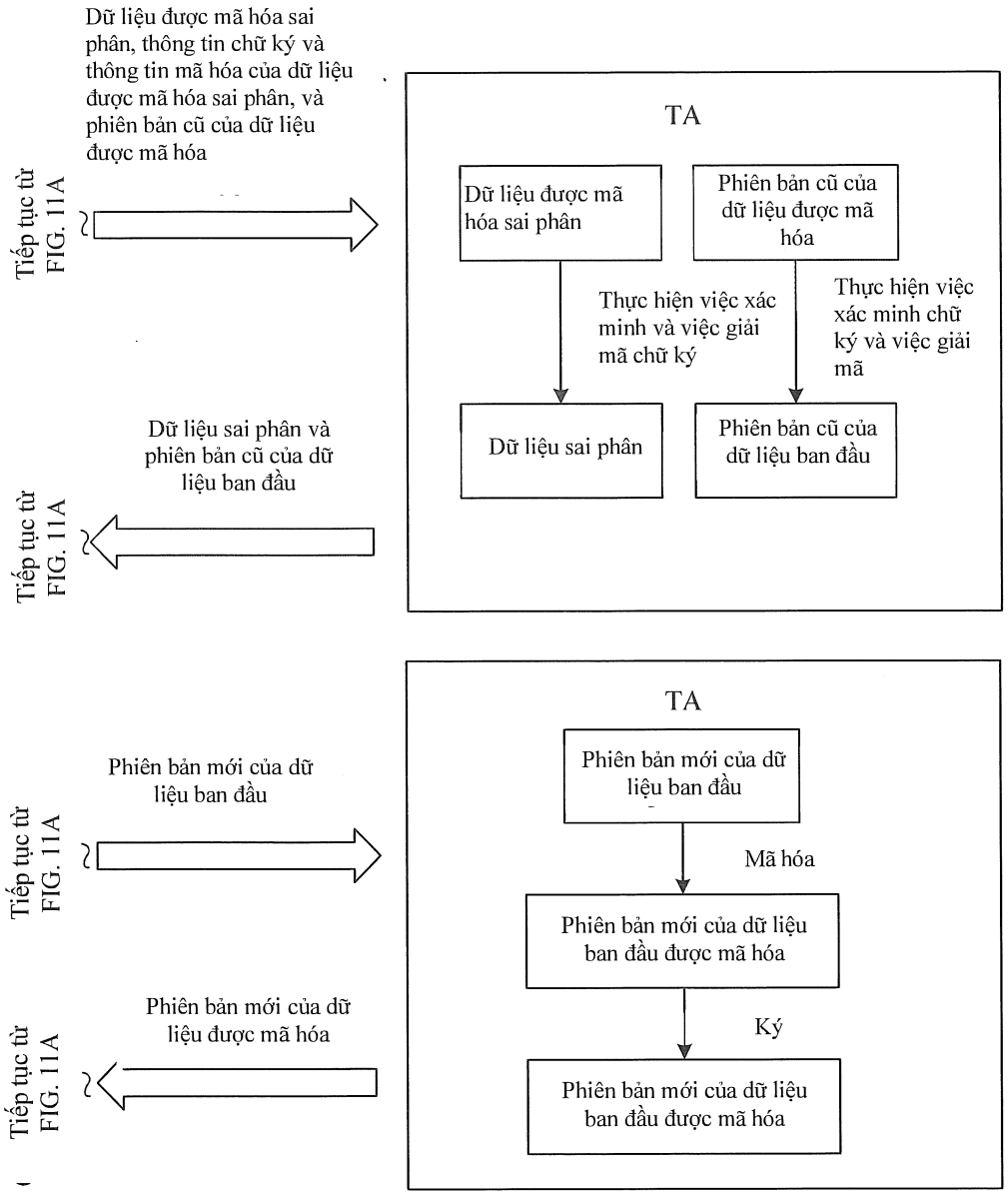

Dữ liệu được mã hóa sai phân, thông tin chữ ký và thông tin mã hóa của dữ liệu được mã hóa sai phân, và phiên bản cũ của dữ liệu được mã hóa
Tiếp tục từ FIG. 11A
TA
Dữ liệu được mã hóa sai phân
Phiên bản cũ của dữ liệu được mã hóa
Thực hiện việc xác minh và việc giải mã chữ ký
Thực hiện việc xác minh chữ ký và việc giải mã
Dữ liệu sai phân
Phiên bản cũ của dữ liệu ban đầu
Dữ liệu sai phân và phiên bản cũ của dữ liệu ban đầu
Tiếp tục từ FIG. 11A
Phiên bản mới của dữ liệu ban đầu
Tiếp tục từ FIG. 11A
TA
Phiên bản mới của dữ liệu ban đầu
Mã hóa
Phiên bản mới của dữ liệu ban đầu được mã hóa
Ký
Phiên bản mới của dữ liệu ban đầu được mã hóa
Phiên bản mới của dữ liệu được mã hóa
Tiếp tục từ FIG. 11A

FIG. 11B

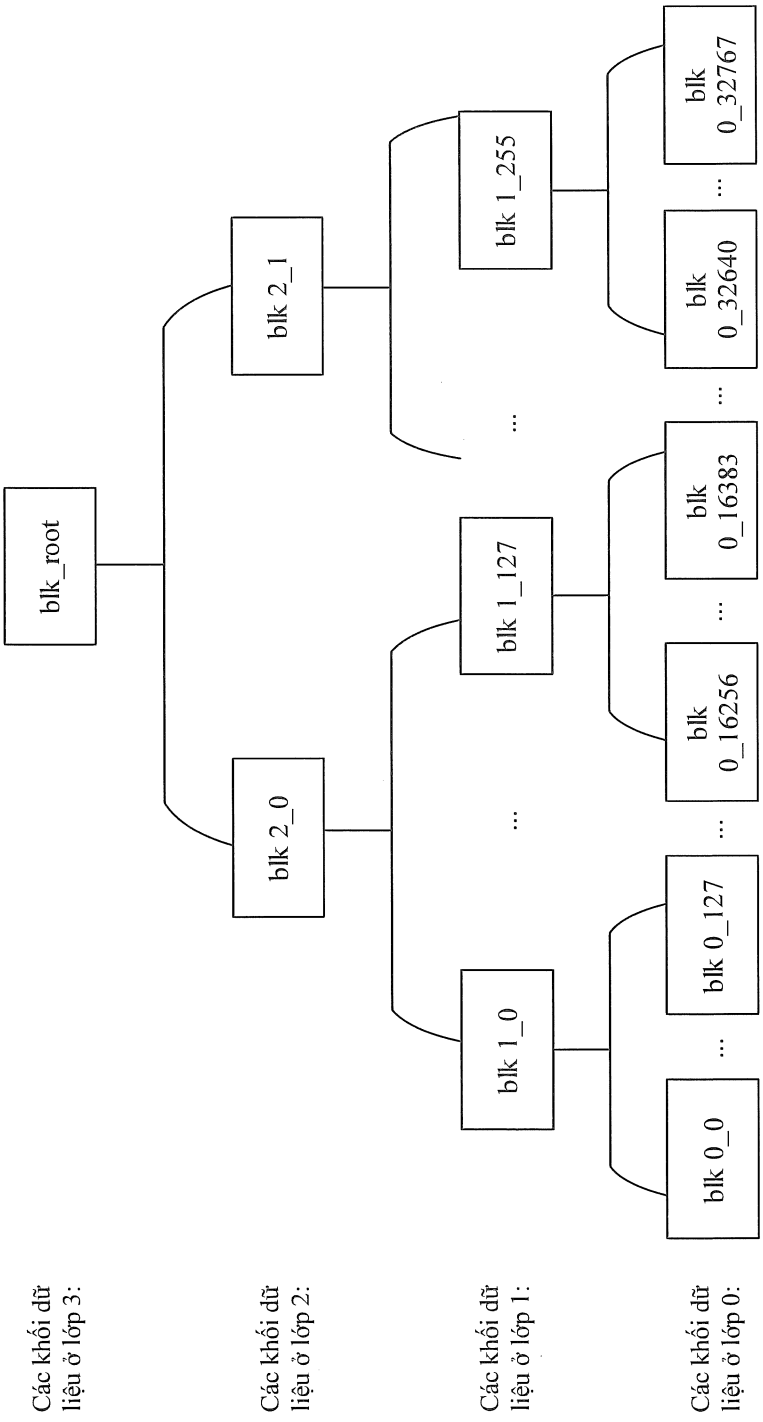
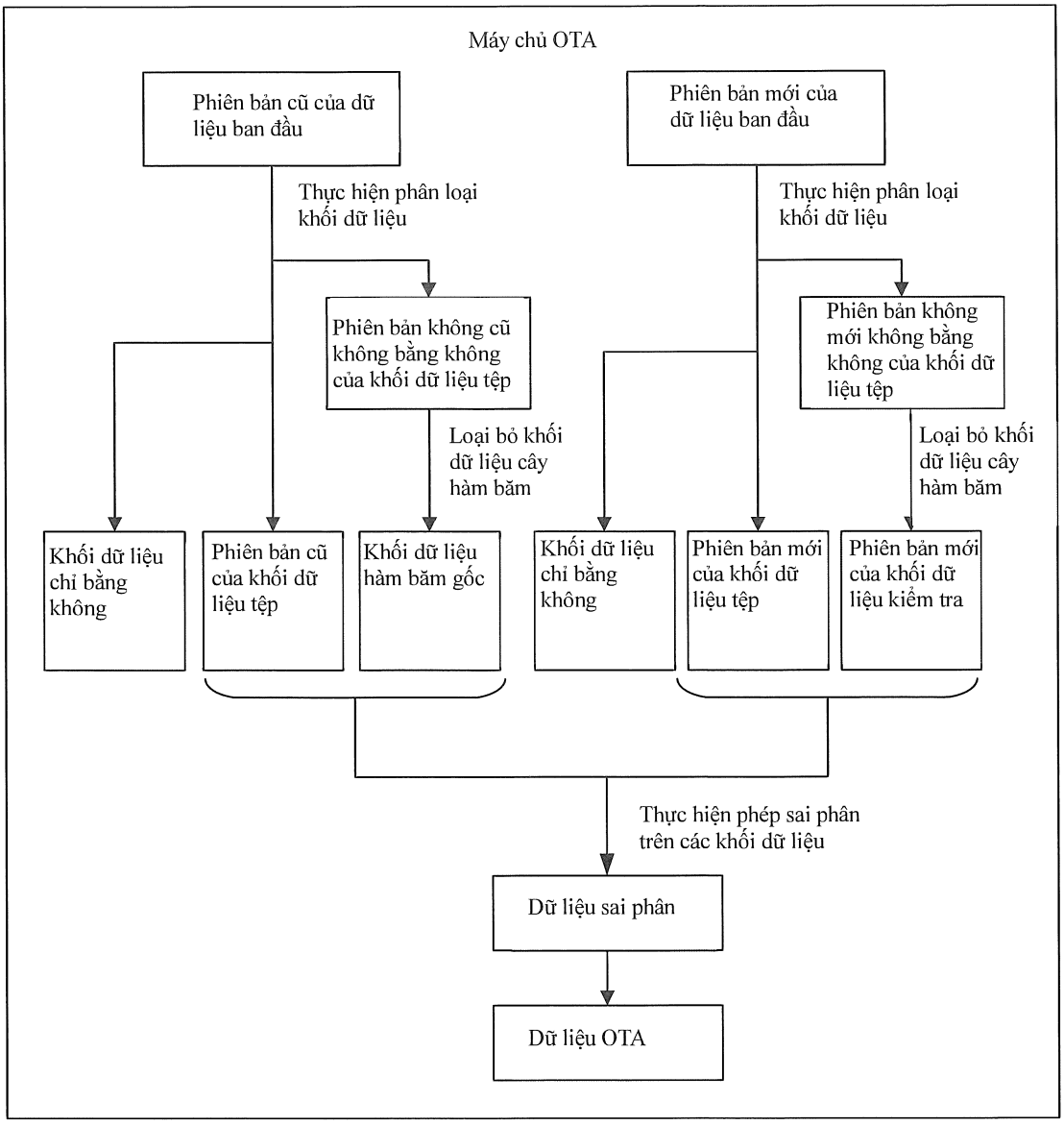

Các khối dữ liệu ở lớp 3:
Các khối dữ liệu ở lớp 2:
Các khối dữ liệu ở lớp 1:
Các khối dữ liệu ở lớp 0:
blk_root
blk 2_0
blk 2_1
blk 1_0
blk 1_127
blk 1_255
blk 0_0
blk 0_127
blk 0_16256
blk 0_16383
blk 0_32640
blk 0_32767
...

FIG. 12



~

Xem tiếp
FIG. 13B

FIG. 13A

Tiếp tục từ
FIG. 13A

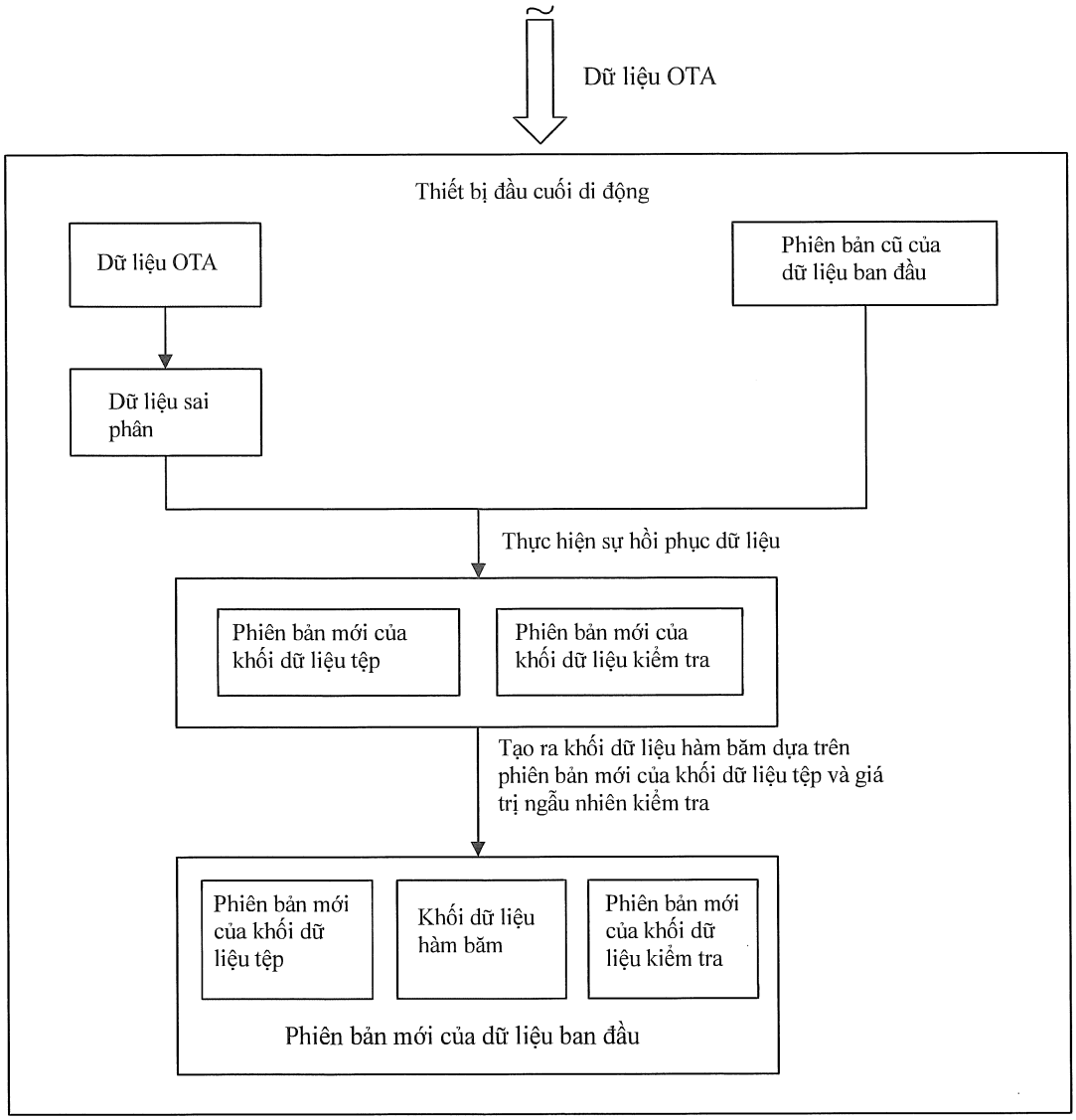


FIG. 13B