SỬ DỤNG THUẬT TOÁN HẠNG NHẸ AEGIS, CLOC CÓ XÁC THỰC ĐỂ BẢO MẬT DỮ LIỆU CAMERA TRONG ĐIỆN THOẠI ANDROID

Nguyễn Như Chiến*, Nguyễn Đức Công, Cầm Văn Dũng

Tóm tắt: *Trong bài báo này, nhóm tác giả giới thiệu về ứng dụng camera trong điện thoại android và hai thuật toán có xác thực hạng nhẹ. Sau đó, nhóm tác giả xây dựng trình ứng dụng bảo mật dữ liệu camera bằng việc thực thi thuật toán hạng nhẹ trên điện thoại android từ việc định nghĩa hàm cập nhật trạng thái, quá trình khởi tạo, xử lý dữ liệu liên kết, hàm mã hóa, quá trình tạo thẻ xác thực, quá trình mã hóa và xác thực tiếp theo là việc thực hiện cài đặt thuật toán và tạo project, cuối cùng đưa ra kết quả minh chứng. Công bố này hỗ trợ sử dụng thuật toán hạng nhẹ aegis, cloc có xác thực để bảo mật dữ liệu camera trong điện thoại android.*

Từ khóa: Hệ thống nhúng; Hệ mã có xác thực; AEGIS; CLOC.

1. GIỚI THIỆU

Trong thời đại công nghệ thông tin bùng nổ như hiện nay, bài toán bảo mật thông tin dữ liệu người dùng ngày càng trở nên cần thiết và cấp bách. Bảo mật thông tin thường đòi hỏi đảm bảo cả tính bí mật và xác thực. Có 2 cách tiếp cận để thực hiện điều này. Cách thứ nhất là mã hóa với các thuật toán mã khối hoặc mã dòng để đảm bảo tính bí mật và sử dụng các thuật toán mã xác thực thông báo MAC (Message Authentication Code) để thực hiện xác thực. Cách tiếp cận thứ hai là xây dựng các thuật toán mã hóa có xác thực nhằm cung cấp tính bí mật, xác thực và toàn vẹn cho thông báo trong một bước duy nhất [1]. Cách tiếp cận này đang là hướng đi mới trong khoa học mật mã.

Cùng với nhu cầu kết nối và trao đổi thông tin mạnh mẽ trên điện thoại thông minh, kéo theo các hiểm họa về mất mát và rò rỉ thông tin của người dùng. Đặc biệt, đối với các hệ điều hành mã nguồn mở như Android [2], cho phép bên thứ ba có quyền sửa đổi mã nguồn theo yêu cầu của họ. Bên cạnh đó, các ứng dụng di động (*Apps*) trao đổi dữ liệu nhạy cảm (thông tin cá nhân, tài khoản email,…) với nhau ngày càng tăng, đã đặt ra yêu cầu phải bảo vệ những dữ liệu này tránh các truy cập trái phép. Một trong số các giải pháp được sử dụng phổ biến nhất là mã hóa dữ liệu. Hơn nữa, điện thoại thông minh cũng phải đáp ứng được nhu cầu trao đổi thông tin với những thiết bị IoT khác có tài nguyên hạn chế, đồng thời vẫn đảm bảo được độ an toàn cần thiết cho kênh truyền [3]. Chính vì vậy, các thuật toán mật mã hạng nhẹ có xác thực [4] là giải pháp hợp lý cho yêu cầu này. Mật mã có xác thực (*AE - authenticated encryption*) hay hệ mật xác thực kèm dữ liệu liên kết (*AEAD - authenticated encryption with associated data*) [6] là một dạng của hệ mật khóa đối xứng đảm bảo tính bí mật, toàn vẹn và xác thực dữ liệu theo từng bước. Nhãn xác thực này cho phép phát hiện các nỗ lực tấn công giả mạo. Trong bài báo này, sẽ thực hiện nhúng và đánh giá hiệu suất của hệ mật AEGIS [5] và CLOC [7] thông qua trình ứng dụng bảo mật dữ liệu camera chạy trên điện thoại Android. Vì những lý do nêu trên mục đích của bài báo này là đưa ra "*sử dụng thuật toán hạng nhẹ Aegis, Cloc có xác thực để bảo mật dữ liệu camera trong điện thoại android*".

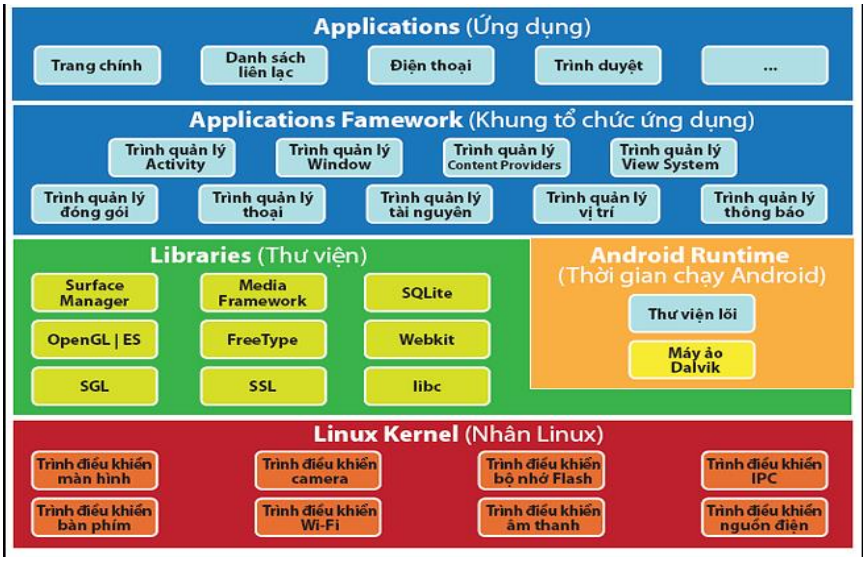
2. CAMERA TRONG ĐIỆN THOẠI ANDROID VÀ HỆ MÃ CÓ XÁC THỰC

Ứng dụng camera trong điện thoại Android

Hệ điều hành Android [10] bao gồm có các thành phần như hình dưới đây:

Các Apps (Facebook, Zalo,…) chạy trên hệ điều hành Android được cài đặt trên tầng ứng dụng. Tầng này cung cấp các phần mềm có các chức năng mặc định của thiết bị Mobile, hỗ trợ phát triển ứng dụng trên Eclipse IDE hoặc Android Studio IDE, và sử dụng ARM Emulator trên QEMU *(Quick EMUlator)*. Ứng dụng camera trong điện thoại Android có thể được hiểu là các

ứng dụng liên quan đến chức năng camera của thiết bị Android, gồm cả ứng dụng máy ảnh được tích hợp sẵn. Các ứng dụng camera trong điện thoại ngoài tính năng như lưu trữ dữ liệu, còn có chức năng trao đổi trực tiếp giữa các người dùng đảm bảo tính bí mật, xác thực và toàn vẹn là chức năng cần thiết và chủ đạo của các ứng dụng camera.



Hình 1. *Kiến trúc của hệ điều hành Android.*

Hệ mã có xác thực Aegis và Cloc

Là hệ mật khóa đối xứng bảo đảm tính bí mật, tính toàn vẹn, và tính xác thực dữ liệu theo từng bước. Trong đó, phép mã hóa sẽ được kết hợp với khối tạo nhãn, còn ở phép giải mã sẽ tiến hành kiểm tra nhãn nhận được. Hệ mã có xác thực hạng nhẹ Aegis và Cloc được trình bày [8, 9].

3. THỰC NGHIỆM VÀ KIỂM CHỨNG KẾT QUẢ

3.1. Xây dựng trình ứng dụng bảo mật dữ liệu camera

Sử dụng Android Studio làm công cụ xây dựng ứng dụng. Android Studio là môi trường phát triển tích hợp (IDE) chính thức để phát triển ứng dụng Android, dựa trên IntelliJ IDEA. Android Studio còn cung cấp nhiều tính năng hơn nữa để nâng cao năng suất của người lập trình khi xây dựng các ứng dụng Android [2].

❖ **Chuẩn bị cấu hình cài đặt:**

- Phần cứng được thực thi trên điện thoại Huawei Nova 2i; Chip HiSilicon Kirin 659; Ram 4GB với hệ điều hành Android 8.0; Ngôn ngữ lập trình: Kotlin. Trong bài viết này nhóm tác giả mô tả AEGIS-128, với khóa 128 bit.

Định nghĩa hàm cập nhật trạng thái: Cập nhật 80 byte trạng thái S_i với một khối thông báo 16 byte (m_i) [5, 7]:

Thuật toán 1 (Hàm cập nhật trạng thái): - **Input:** S (State); - **Ouput:** S (New state)

Begin 1. $S_4 \leftarrow AESRound(S_3, S_4)$; $S_3 \leftarrow AESRound(S_2, S_3)$; $S_2 \leftarrow AESRound(S_1, S_2)$;
$S_1 \leftarrow AESRound(S_0, S_1)$; $S_0 \leftarrow AESRound(S_4, S_0)$;

End 2. Return S

❖ **Quá trình khởi tạo:** Việc khởi tạo AEGIS-128 bao gồm nạp khóa và Nonce vào trạng thái S, các bước được thực hiện như sau [5, 7]:

Thuật toán 2 (Quá trình khởi tạo): - **Input:** K (Key), N (Nonce); - **Ouput:** S

Begin 1. $S \leftarrow (K, N, const)$;
2. *for (i= 0; i<10; i++)*
{StateUpdate (); $K \leftarrow K \oplus N$; $S_0 \leftarrow K$;*}*
End 3. Return S

❖ **Hàm mã hóa:** Sau khi xử lý dữ liệu liên kết, tại mỗi bước của quá trình mã hóa, một khối bản rõ 16 byte P_i được sử dụng để cập nhật trạng thái, và P_i được mã hóa thành C_i [5, 7].

Thuật toán 3 (Hàm mã hóa): - **Input:** S, P (Plaintext); - **Ouput:** C

Begin 1. *for (i=0; i+16<= Plen; i+=16){*
2. $C_i = S_1 \oplus S_4 \oplus (S_2 \,\&\, S_3)$; $S_0 \leftarrow P_i \oplus S_0$;
StateUpdate();}
End 3. Return C

❖ **Quá trình tạo thẻ xác thực:** Sau khi mã hóa tất cả các khối bản rõ, chúng ta tạo thẻ xác thực bằng bảy bước khác. Độ dài của dữ liệu liên quan và độ dài của thông báo được sử dụng để cập nhật trạng thái [5, 7].

Thuật toán 4 (Hàm tạo thẻ xác thực): - **Input:** S, ADlen, Plen; - **Ouput:** T

Begin 1. $tmp \leftarrow S_3 \oplus (ADlen \,||\, Plen)$; *for (i=0;i<7;i++) {StateUpdate();*
$S_0 \leftarrow tmp \oplus S_0$; *}* $S_4 \leftarrow S_4 \oplus S_3 \oplus S_2 \oplus S_1 \oplus S_0$;
End 2. Return T

❖ **Quá trình giải mã và xác thực:** Quá trình giải mã, một khối bản mã 16 byte C_i được sử dụng để cập nhật trạng thái, và C_i được giải mã thành P_i [5, 7].

Thuật toán 5 (Hàm giải mã): - **Input:** S, C (Ciphertext); - **Ouput:** P

Begin 1. *for (i=0;i+16<=Plen;i+=16){*
$P_i \leftarrow C_i \oplus S_1 \oplus S_4 \oplus (S_2 \,\&\, S_3)$; *StateUpdate();* $S_0 \leftarrow S_0 \oplus P_i$; *}*
End 2. Return P

3.2. Quá trình thực thi thuật toán

❖ **Thực thi thuật toán trên điện thoại Android:** Để phục vụ lệnh mã hóa, phương thức mã hóa đã khai báo (*AEGISEncryption*) phải được tạo các như sau:

Thuật toán 6 (Mã hóa ảnh): - **Input:** K, N, AD, file.jpg; - **Ouput:** file.dat

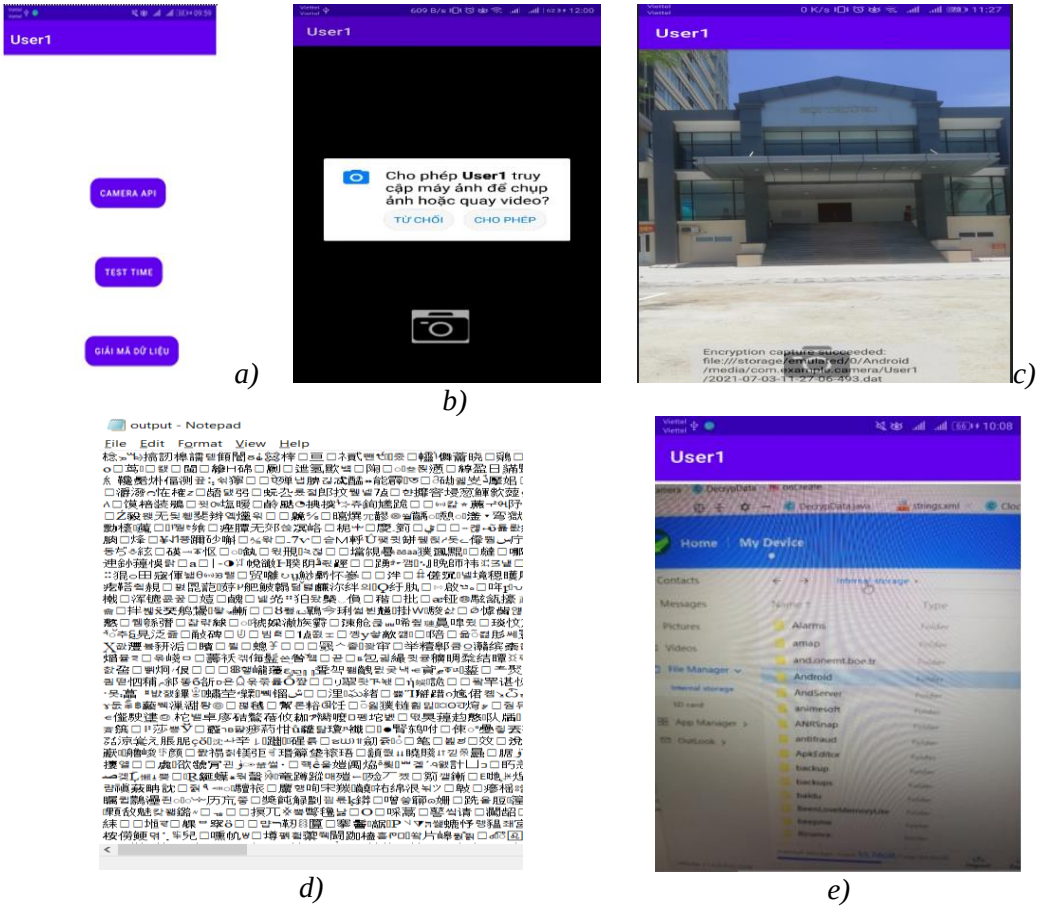
Begin 1. $P \leftarrow File.dat(tobyte.\text{array})$; *Initialization(K, N);*
ProcessAD(AD); $C \leftarrow Encryp(P)$;
End 2. Return $File.dat \leftarrow C$.

- Việc giải mã bắt đầu với việc khởi tạo và xử lý dữ liệu đã được xác thực. Quá trình giải mã hoàn tất, khối 16 byte cuối cùng sẽ được cộng xor với nhãn xác thực *T*, quá trình diễn ra như sau:

Thuật toán 7 (Giải mã ảnh): - **Input:** K, N, AD, file.dat; - **Ouput:** file.jpg or False

Begin 1. $C \leftarrow File.dat(tobyte.\text{array})$; *Initialization(K,N);*
ProcessAD(AD); $P \leftarrow Decryp(C)$;
$T \leftarrow Tag\_genration$; $Check \leftarrow T \oplus C_{Plen}$;
End 2. *If (Check == 0) Return* $File.jpg \leftarrow P$; *Else Return False;*

Kết quả được hiển thị như hình 2. Hình 2 thể hiện dữ liệu sau khi qua giao diện app (hình 2a); chọn camera API ứng dụng từ việc chấp nhận cấp quyền người dùng (hình 2b); khi chụp hình thu được từ camera và mã hóa, sau đó file mã được lưu vào bộ nhớ dưới dạng file.dat (hình 2c); bản mã thu được sau khi mã (hình 2d); quá trình giải mã sẽ được hiển thị trên ImageView và được lưu lại trong đường dẫn lưu mặc định của ứng dụng (hình 2e). Tên các file mã/rõ được gắn thêm tem thời gian để đảm bảo tính duy nhất cho mỗi tên file.



Hình 2. *Các trạng thái xử lý bảo mật dữ liệu camera trên điện thoại Android.*

Dưới đây là bảng thống kê kết quả dung lượng bộ nhớ bị chiếm dụng của từng thuật toán khi nhúng vào Javacard bảng 1 và bảng thời gian mã hóa các khối dữ liệu của Aegis và Cloc đo được trên thiết bị Android bảng 2 và bảng 3.

Bảng 1. *Thống kê kết quả dung lượng bộ nhớ bị chiếm dụng của từng thuật toán.*

Hệ mật	RAM (Byte)	EEPROM (Byte)	Kích thước hệ mật (Byte)
AEGIS	468	4176	12623
CLOC	832	1247	10149

Bảng 2. *Thời gian mã hóa các khối dữ liệu của AEGIS.*

Size (KB)	10	15	20	25	30
Time (ms)	37	60	71	77	108

Bảng 3. *Thời gian mã hóa các dữ liệu của CLOC*

Size (KB)	10	15	20	25	30
Time (ms)	47	72	88	118	126

4. KẾT LUẬN

Trong bối cảnh nhu cầu về mật mã hạng nhẹ ngày càng trở nên cần thiết cho sự phát triển của IoT, cùng với việc xuất hiện những điểm yếu trong thuật toán AES-GCM, đòi hỏi sự ra đời của các hệ mật mới nhằm giải quyết vấn đề này. Thông qua nghiên cứu và thực thi nhúng hai hệ mật mã hạng nhẹ có xác thực: AEGIS và CLOC trong App bảo mật dữ liệu camera chạy trong điện thoại Andoroi. Kết quả ban đầu cho thấy, App nhúng hai hệ mật này chạy thông suốt trong điều kiện tuân thủ các cấu hình cài đặt.

Tiến hành đo và đánh giá hiệu quả của 2 thuật toán dựa vào thời gian thực thi, AEGIS vượt trội về mặt tốc độ so với CLOC. Tuy nhiên, các thiết bị điện thoại thông minh có cấu hình phần cứng như hiện nay, các hệ mã xác thực hạng nhẹ chưa thực sự có nhiều ý nghĩa trong việc tối ưu bộ nhớ, do kích thước bộ nhớ mà các thuật toán sử dụng là không đáng kể so với tài nguyên của thiết bị.

Trong tương lai, nhóm hướng đến xây dựng ứng dụng mã/giải mã thông tin video chạy trên nền tảng Android, đóng gói chương trình thành file *.apk cho phép cài đặt và bảo mật truyền video theo thời gian thực giữa các điện thoại Android.

TÀI LIỆU THAM KHẢO

[1]. CAESAR: "Competition for Authenticated Encryption: Security, Applicability, and Robustness". https://competitions.cr.yp.to/caesar-submissions.html.

[2]. John Wiley & Sons, "*Java Programming for Android Developers for Dummies*", 2014.

[3]. Masanobu Katagi and Shiho Moriai, "*Lightweight Cryptography for the Internet of Things*".

[4]. "*Lightweight Cryptography Program of NIST*", [Online]. Available: https://csrc.nist.gov/projects/lightweight-cryptography.

[5]. Hongjun Wu, Bart Preneel, "*AEGIS: A Fast Authenticated Encryption Algorithm*", 2016.

[6]. D. Bhattacharjee and A. Chattopadhyay, "*Efficient Hardware Accelerator for AEGIS-128 Authenticated Encryption. International Conference on Information Security and Cryptology*", 2015.

[7]. Tetsu Iwata, Kazuhiko Minematsu, Jian Guo, and Sumio Morioka, "*CLOC: Authenticated Encryption for Short Input*", The Pre-proceedings of FSE 2014.

[8]. Đinh Tiến Thành, Nguyễn Đức Công, Nguyễn Như Chiến, "*Một phương pháp tuyển chọn hệ mã có xác thực hạng nhẹ dùng cho thẻ thông minh*", Tạp chí nghiên cứu Khoa học và Công nghệ Quân sự, Số 59 02/2019.

[9]. Nguyễn Như Chiến, "*Hệ mật AEGIS, CLOC và thực thi hệ mật trên thẻ thông minh*", Tạp chí An toàn Thông tin – Ban Cơ yếu Chính phủ, Tạp chí in số 2 Quý 3/2021.

[10]. https://viblo.asia/p/kien-truc-cua-he-dieu-hanh-android-PaLGDYdaelX

ABSTRACT

USING LIGHTWEIGHT AUTHENTICATED ENCRYPTION ALGORITHM AEGIS, CLOC TO ENCRYPT CAMERA DATA ON ANDROID PHONES

This paper addresses the camera application on Android phones and two lightweight authenticated encryption algorithms. We construct an application to encrypt camera data by implementing a lightweight algorithm on Android phones, includes the state update function, the initialization, the authenticated data processing, the encryption, the generation of the authentication tag and the decryption and verification. Then, we install the algorithm and create a project, finally we present the experimental results. Our contribution supports using the lightweight authenticated encryption algorithm AEGIS, CLOC for the secure protection of camera data on Android phones.

Keywords: Embedded systems; Authenticated encryption algorithm; AEGIS; CLOC.

Nhận bài ngày 22 tháng 7 năm 2021
Hoàn thiện ngày 26 tháng 8 năm 2021
Chấp nhận đăng ngày 10 tháng 10 năm 2021

Địa chỉ: Học viện Kỹ thuật Mật mã – Ban Cơ yếu Chính phủ.
*Email: chienct1102@gmail.com.