NGHIÊN CỨU MỘT SỐ HỆ MẬT MÃ NHẸ VÀ ỨNG DỤNG TRONG IoT

Lê Phê Đô, Mai Mạnh Trừng, Lê Trung Thực, Nguyễn Thị Hằng, Vương Thị Hạnh, Nguyễn Khắc Hưng, Đinh Thị Thúy, Lê Thị Len[1]

Tóm tắt: *Người ta ước tính đến năm 2020 sẽ có hơn 50 tỷ thiết bị kết nối internet, nghĩa là mỗi người trên trái đất trung bình sẽ có 6,6 đồ vật trực tuyến. Trái đất sẽ được che phủ bởi hàng triệu cảm biến thu thập thông tin và tải lên internet. Để đảm bảo các kết nối được an ninh và an toàn các thiết bị đó cần có các hệ mật vừa có độ mật cần thiết, tiêu tốn ít năng lượng, bộ nhớ và các cổng logic. Đó là các hệ mật mã nhẹ, gồm mã khối hạng nhẹ, mã dòng hạng nhẹ và các mã xác thực hạng nhẹ. Trong báo cáo này chúng tôi giới thiệu một số hệ mật trong mật mã nhẹ, đưa ra những điểm mạnh và điểm yếu của chúng. Các hệ mã khối hạng nhẹ được nghiên cứu gồm Klein, Led, Present, Mini – AES, Mcrypyon và Katan. Hệ mã dòng được chúng tôi giới thiệu là Grain. Kết quả có thể dùng làm tài liệu tham khảo cho các nhà chuyên môn về mật mã nhẹ và IoT.*

Từ khóa: Mật mã nhẹ, mã khối, mã dòng, IoT, Present, Grain, Độ trễ, Hiệu suất, Độ an toàn,…

1. MỞ ĐẦU

Với các thiết bị có tài nguyên hạn chế thì các thuật toán mật mã thông thường là quá lớn, quá chậm và quá tốn năng lượng. Các thuật toán mật mã nhẹ khắc phục được những nhược điểm này. Mục tiêu của mật mã nhẹ là một loạt các ứng dụng cho các thiết bị hiện đại, như các thiết bị đo thông minh, hệ thống an ninh xe, hệ thống giám sát bệnh nhân không dây, hệ thống giao thông thông minh (ITS) và Internet of Things (IoT), …

Trong thiết kế của mật mã hạng nhẹ sự cân bằng giữa chi phí, an ninh và hiệu suất phải được đảm bảo. Vì các mã khối, độ dài khóa đưa ra sự thỏa hiệp giữa độ an toàn và giá thành, trong khi đó số vòng đưa ra thỏa hiệp giữa hiệu suất và độ an toàn. Thông thường, ta có thể dễ tối ưu hóa được hai tiêu chí bất kỳ trong ba tiêu chí trên, nhưng việc tối ưu hóa cả ba mục tiêu là việc rất khó. Bên cạnh đó, cài đặt bằng phần cứng có hiệu suất cao cũng cần tính tới giải pháp để tránh các tấn công kênh kề. Điều này thường dẫn tới các yêu cầu về diện tích cao, đồng nghĩa với chi phí cao.

Các yêu cầu thiết kế và mật mã hạng nhẹ cần:

Về độ an toàn, mục tiêu xây dựng các hệ mã hạng nhẹ là thiết kế một hệ mật không quá yếu (và không với mục đích thay thế các thuật toán mã truyền thống khác), nhưng phải đủ an toàn (tất nhiên không thể kháng lại được các đối phương có đủ mọi điều kiện), chi phí (cài đặt, sản xuất) thấp và một yêu cầu quan trọng đối với các thiết bị kiểu này là tính gọn nhẹ "on-the-fly". Tóm lại, cần xây dựng một hệ mật không phải tốt nhất, mà phải cân bằng giữa giá thành, hiệu suất và độ an toàn.

Về hiệu quả trong cài đặt, thường được đánh giá qua các độ đo sau: diện tích bề mặt (Area), Số chu kỳ xung nhịp (cycles), Thời gian, Thông lượng (throughout), Nguồn (power), Năng lượng (energy), Dòng điện (current). Tính hiệu quả là tỷ lệ thông lượng với diện tích, được dùng làm độ đo cho tính hiệu quả phần cứng.

- **Diện tích bề mặt (Area):** Có thể tính bằng micro m^2 nhưng giá trì này phụ thuộc vào công nghệ chế tạo và thư viện chuẩn. Diện tích tính theo GE được tính bằng cách chia diện tích theo micro m^2 cho S cổng NAND 2 đầu vào.
- **Số chu kỳ xung nhịp (cycles):** là số chu kỳ xung nhịp cần để tính toán và đọc dữ liệu ra.
- **Thời gian**: Lượng thời gian cần thiết cho một phép tính cụ thể có thể được tính bằng cách chia số chu kỳ xung nhịp cho tần số hoạt động t = (số chu kỳ xung nhịp)/tần số. Đơn vị tính theo mi-ni giây (ms).
- **Thông lượng (throughtout):** Là số các bit đầu ra chia cho 1 lượng thời gian nào đó. Đơn vị [bps]

- **Nguồn (power):** Tiêu thụ nguồn có thể được ước lượng ở mức cổng thông qua bộ biên dịch cài đặt. Đơn vị thường Micro walt. Chú ý việc ước lượng tiêu thụ ở mức transitor là chính xác hơn, nhưng điều này sẽ yêu cầu nhiều bước hơn khi thiết kế.
- **Năng lượng (energy):** Tiêu thụ năng lượng được định nghĩa là tiêu thụ nguồn qua 1 khoảng thời gian cụ thể. Nó thường được tính toán bằng cách nhân tiêu thụ nguồn với thời gian cần cho phép tính đó, đơn vị Joule trên bit.
- **Dòng điện(current):** Là tiêu thụ nguồn chia cho điện áp thông thường.
- **Tính hiệu quả cài đặt:** eff = (diện tích)/ thông lượng

2. MỘT SỐ HỆ MẬT MÃ KHỐI HẠNG NHẸ

2.1. Giới thiệu các thuật toán mã khối hạng nhẹ

Mã khối hạng nhẹ là một nhóm thuộc mật mã nhẹ sử dụng trong an toàn thông tin, ở đó thuật toán mã hóa sử dụng đầu vào là các khối B-bit và khóa là K-bit.

Một số hệ mật mã khối hạng nhẹ tiêu biểu thường được sử dụng trên thế giới hiện nay:

Hệ mật	**Kích thước khối tin**	**Độ dài khóa**	**Số vòng mã hóa**
KLEIN	64 bits	64 – 80 – 96 bits	12 – 16 – 20
LED	64 bits	64 - 128 bits	32 - 48
PRESENT	64 bits	80 - 128 bits	31
MINI-AES	64 bits	64 bits	10
MCRYPTON	64 bits	64 – 96 - 128 bits	12
KATAN	32 – 48 – 64 bits	80 bits	

2.2. Đánh giá các thuật toán

Chúng tôi thực hiện đánh giá thuật toán mã hóa qua các tiêu chí: độ trễ xử lý, số lượng cổng tương đương, năng lượng tiêu thụ, độ an toàn.

Độ trễ xử lý:

Định nghĩa 1. Độ trễ xử lý thuật toán [3]

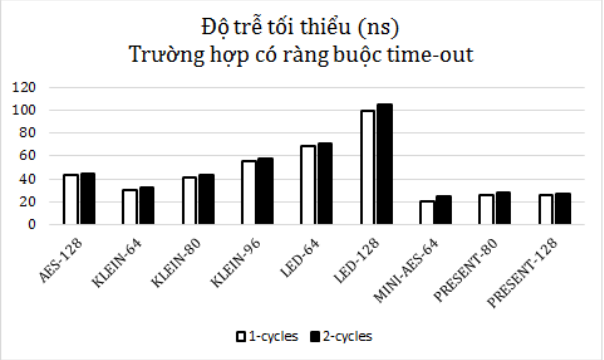
Độ trễ xử lý thuật toán đại diện cho khoảng thời gian để thuật toán hoàn thiện xử lý một nhiệm vụ. Trong bài báo này, chúng tôi sử dụng nó là thước đo thời gian mã hóa một khối bản rõ xác định. Độ trễ xử lý thuật toán L được tính bởi công thức:

$$L = N.t_{cp}$$

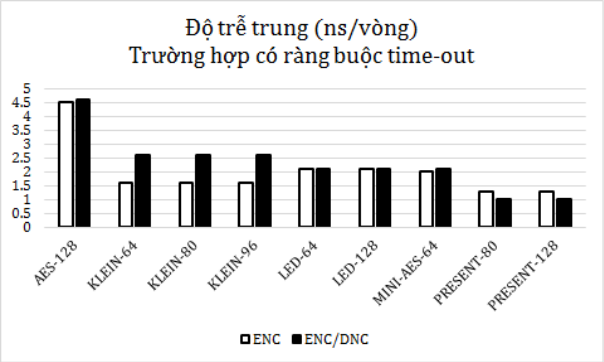
Trong đó,

- N: số xung nhịp cần để thực hiện một chu kỳ mã hóa
- t_{cp}: độ trễ tối đa thời gian thực hiện một chu kỳ mã hóa
- Đơn vị của L được tính bằng nano giây (ns): $1ns = 10^{-9}s$.

Chúng tôi tổng hợp các kết quả ước tính trong sản xuất của công ty NXP Semiconductors, một đơn vị sản xuất sản phẩm về vi mạch điện tử tích hợp tại Bỉ. Từ đó có thể trực quan đánh giá định lượng độ trễ của các thuật toán mã hóa, kết quả thực nghiệm được xét trong 2 trường hợp: (1) Không ràng buộc về thời gian time-out và (2) Có ràng buộc về thời gian time-out.



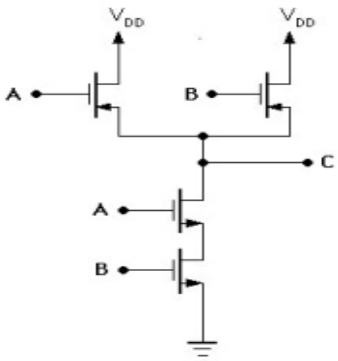
Hình 1. Ước tính độ trễ [3, 4]



Hình 2. Ước tính độ trễ trung bình [3, 4]

Số lượng cổng tương đương:

Định nghĩa 2. Cổng tương đương - *Một cổng tương đương được mô phỏng bằng diện tích vật lý mà một cổng logic NAND hai đầu vào chiếm trong vi mạch điện tử.*



Đơn vị của cổng tương đương là GE (Gate equivalence), 1kGE = 1000GE. Một số các phép toán logic tương đương tiêu biểu trong thuật toán mật mã: AND, NAND, OR, XOR, NOR, NOT.

Bằng những thực nghiệm, các kỹ sư nghiên cứu của NXP đã đưa ra được các kết quả ước tính khi đo trên cùng một chu kỳ mã hóa đối với một số thuật toán: KATAN (460GE), PRESENT (1kGE), LED (700GE), SIMON (520GE), PICCOLO (700GE/180ns), KLEIN (700GE/130ns).

Tiêu thụ năng lượng:

Định nghĩa 3. Mức tiêu thụ năng lượng

Kết quả cho mức tiêu thụ năng lượng trung bình được ước tính dựa trên hoạt động chuyển mạch của mạch. Trong bài báo chúng tôi tin tổng hợp các ước tính của một số nghiên cứu đáng tin cậy.

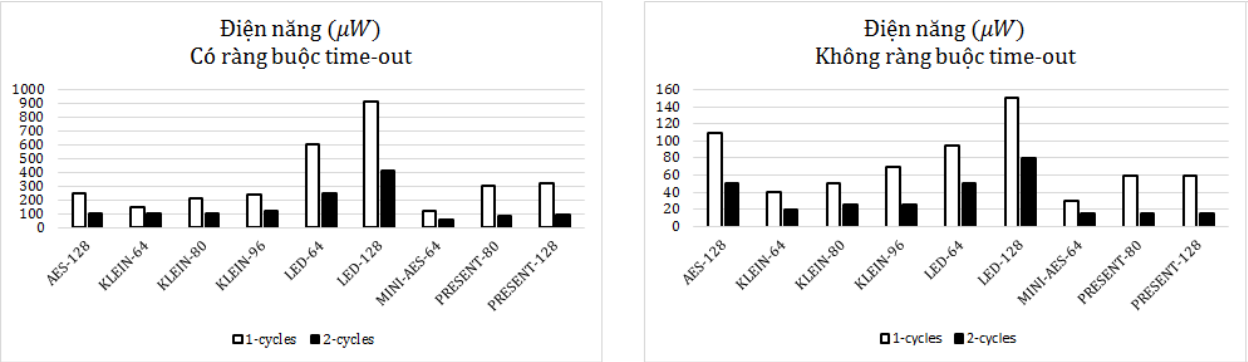
Mức tiêu thụ năng lượng được ước tính dựa trên công thức [1] sau:

$$E = \frac{P.L}{B} = \frac{P.N.t_{cp}}{B}$$

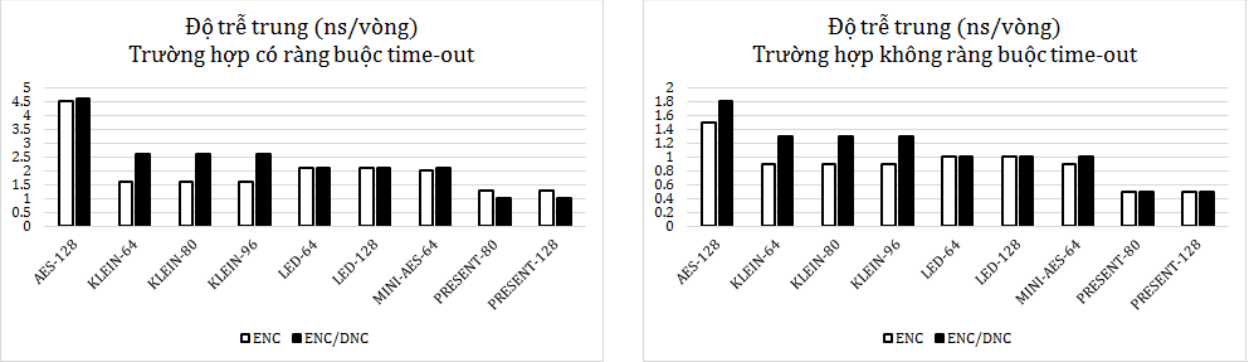
Trong đó,

- P: điện năng tiêu thụ trung bình

- N: số xung nhịp cần để thực hiện một chu kỳ mã hóa
- t_{cp}: độ trễ tối đa thời gian thực hiện một chu kỳ mã hóa
- B: kích thước bản tin



Hình 3. Ước tính điện năng tiêu thụ [1, 2, 3]



Hình 4. Ước tính năng lượng tiêu thụ [1, 2, 3]

Độ an toàn:

Định nghĩa 4. Khoảng cách tổng biến thiên [5]: Gọi X, Y lần lượt là hai biến ngẫu nhiên trên tập hữu hạn Q. Khoảng cách tổng biến thiên của X và Y được xác định bởi:

$$d^{TV}(X,Y) \triangleq \max_{A \subset Q} |\Pr[X \in A] - \Pr[Y \in A]|.$$

Đại lượng này thường được dùng trong mật mã để phân tích độ an toàn của thuật toán trước các dạng tấn công chung. Có thể hiểu đó là xác suất thành công lớn nhất của việc tấn công ở hai trường hợp: trường hợp lý tưởng và trường hợp thực tế.

Ý tưởng thiết kế thuật toán mới:

Một câu hỏi mà tất cả các nhà thiết kế cần giải quyết trong khi thiết kế bất kỳ mã pháp nào là "*độ an toàn bao nhiêu thì được coi là đủ an toàn*". Do đó, nếu một cơ chế an toàn được triển khai không được sử dụng đầy đủ khả năng của nó sẽ dẫn tới việc lãng phí tài nguyên. Một ví dụ, ta đều biết rằng AES đã được phân tích rộng rãi đối với độ an toàn của nó. Cho đến nay, nó đã được chứng minh kháng lại rất nhiều tấn công. Do đó, thật lý tưởng khi các nhà cung cấp phát triển được thuật toán AES trong các thiết bị của họ. Tuy nhiên, một vấn đề gặp phải đối với AES là nó rất cồng kềnh và cần rất nhiều tài nguyên cho việc cài đặt. Ngoài ra, nó cung cấp độ an toàn nhiều hơn những gì cần thiết cho việc sử dụng [2]. Vì vậy, ta cần thấy rằng để thiết kế một nguyên thủy phù hợp với các hạn chế về tài nguyên của các thiết bị nhỏ và cùng lúc các nguyên thủy này cũng cung cấp độ an toàn đầy đủ cho việc sử dụng. Đây cũng chính là một trong những nguyên nhân chính thúc đẩy mật mã hạng nhẹ phát triển. Bây giờ, ta xem xét khía cạnh kỹ thuật của thiết kế mã khối, sau khi quyết định chọn lựa các tham số đầu vào

phù hợp việc tiếp theo mà người thiết kế quan tâm chính là hàm vòng. Đặc biết đối với mã khối hạng nhẹ, hàm vòng phải thật đơn giản đối với cài đặt phần cứng. Một hàm vòng chứa một hàm phi tuyến và một hàm tuyến tính. Hàm phi tuyến được gọi là tầng xáo trộn còn hàm tuyến tính được gọi là tầng khuếch tán. Do vậy, chúng ta dựa vào hai phương pháp quan trọng là xáo trộn và khuếch tán trong việc xây dựng hàm vòng. Mục đích của hai hàm này được phát biểu cụ thể như sau:

- **Xáo trộn (confusion):** Sự phụ thuộc của bản mã đối với bản rõ phải thực phức tạp để gây rắc rối, cảm giác hỗn loạn đối với kẻ thù có ý định phân tích tìm qui luật để phá mã. Quan hệ hàm số của mã-tin là phi tuyến (non-linear).
- **Khuếch tán (diffusion):** Làm khuếch tán những mẫu văn bản mang đặc tính thống kê (gây ra do độ dư ngôn ngữ) lẫn vào toàn bộ văn bản. Nhờ đó tạo ra khó khăn cho kẻ thù trong việc dò phá mã trên cơ sở thống kê các mẫu lặp lại cao. Sự thay đổi của một bit trong một khối bản rõ phải dẫn tới sự thay đổi hoàn toàn trong khối mã tạo ra.

3. HỆ MẬT GRAIN

3.1 Lịch sử

Grain là hệ mật mã dòng được đăng trên eSTREAM bởi Martin Hell, Thomas Johansson và Willi Meier năm 2004 với phiên bản đầu tiên Grain v0 [8]. Sau đó hệ mật này tiếp tục được phát triển thành Grain v1 [7] – là một trong bảy dự án được eSTREAM đưa vào các danh mục đầu tư từ 09/09/2008. Cùng với Grain v1 là một phiên bản mật mã với khóa bí mật 128 bits – Grain-128 [7] cũng được áp dụng rộng rãi hiện nay.

3.2 Mô tả thuật toán

Grain v0

Grain là một hệ mã hóa dòng đồng bộ, các khóa dòng sẽ được tạo một cách độc lập từ bản rõ. Thiết kế của Grain được dựa trên hai thanh ghi dịch chuyển, một thanh ghi dịch hồi tuyến tính (LFSR - linear feedback shift register) và một thanh ghi phản hồi phi tuyến (NFSR - nonlinear feedback shift register). Độ dài các thanh ghi dịch dù là phản hồi tuyến tính hay phản hồi phi tuyến nên là nguyên tố cùng nhau để tránh xuất hiện chu kỳ con khi tạo dãy bit ngẫu nhiên và ô đầu tiên của chúng là chứa bit 1 [13]. Hai thanh ghi này cùng với một hàm đầu ra tạo ra ba khối chính cho thuật toán. Nội dung của LFSR được biểu diễn bằng $s_i, s_{i+1}, \dots, s_{i+79}$ và nội dung của NFSR được mô tả bằng $b_i, b_{i+1}, \dots, b_{i+79}$.

Đa thức nguyên thủy của thanh ghi dịch hồi tuyến tính: $f_0(x) = 1 + x^{18} + x^{29} + x^{42} + x^{57} + x^{67} + x^{80}$

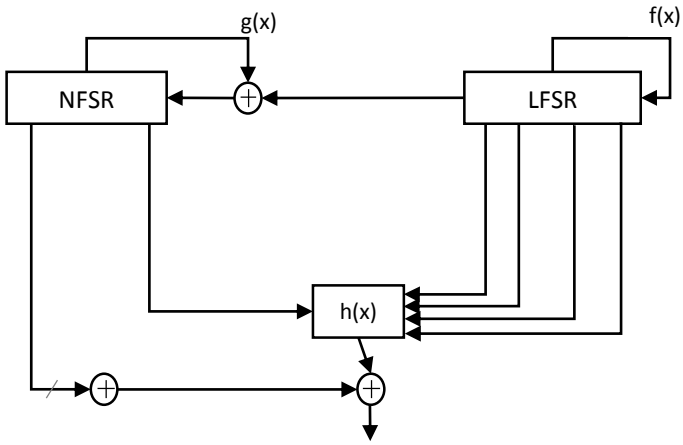
Ta sử dụng một phiên bản cập nhật của LFSR như sau: $s_{i+80} = s_{i+62} + s_{i+51} + s_{i+38} + s_{i+23} + s_{i+13} + s_i$

Hàm của bộ ghi dịch hồi phi tuyến (NFSR) được định nghĩa như sau:

$$\begin{aligned} g_0(x) = 1 + x^{18} + x^{20} + x^{28} + x^{35} + x^{43} + x^{47} + x^{52} + x^{59} + x^{66} + x^{71} + x^{80} + x^{17}x^{20} \\ + x^{43}x^{47} + x^{65}x^{71} + x^{20}x^{28}x^{35} + x^{47}x^{52}x^{59} + x^{17}x^{35}x^{52}x^{71} + x^{20}x^{28}x^{43}x^{47} \\ + x^{17}x^{20}x^{59}x^{65} + x^{17}x^{20}x^{28}x^{35}x^{43} + x^{47}x^{52}x^{59}x^{65}x^{71} + x^{28}x^{35}x^{43}x^{47}x^{52}x^{59} \end{aligned}$$

Loại bỏ những giá trị không cần thiết ta được hàm cập nhật như sau:

$$\begin{aligned} b_{i+80} = {} & s_i + b_{i+62} + b_{i+60} + b_{i+52} + b_{i+45} \\ & + b_{i+37} + b_{i+33} + b_{i+28} + b_{i+21} \\ & + b_{i+14} + b_{i+9} + b_i + b_{i+63}b_{i+60} \\ & + b_{i+37}b_{i+33} + b_{i+15}b_{i+9} \\ & + b_{i+60}b_{i+52}b_{i+45} \\ & + b_{i+33}b_{i+28}b_{i+21} \\ & + b_{i+63}b_{i+45}b_{i+28}b_{i+9} \\ & + b_{i+60}b_{i+52}b_{i+37}b_{i+33} \\ & + b_{i+63}b_{i+60}b_{i+21}b_{i+15} \\ & + b_{i+63}b_{i+60}b_{i+52}b_{i+45}b_{i+37} \\ & + b_{i+33}b_{i+28}b_{i+21}b_{i+15}b_{i+9} \\ & + b_{i+52}b_{i+45}b_{i+37}b_{i+33}b_{i+28}b_{i+21} \end{aligned}$$



Nội dung của hai thanh ghi được thay đổi trạng thái của mã hóa. Từ 5 biến đầu vào, qua hàm logic h(x) được cân bằng với một đầu ra của hàm phi tuyến NFSR.

$$\mathrm{h}_0(\mathrm{x}) = x_1 + x_4 + x_0x_3 + x_2x_3 + x_3x_4 + x_0x_1x_2 + x_0x_2x_3 + x_0x_2x_4 + x_1x_2x_4 + x_2x_3x_4$$

trong đó x_0, x_1, x_2, x_3, x_4 tương ứng với các vị trí $s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}, b_{i+63}$. Đầu ra của hàm này sẽ là ${z^0}_i = \sum_{k\in A} b_{i+k} + \mathrm{h}_0(s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}, b_{i+63})$

Trong đó $A = \{1, 2, 3, 10, 31, 43, 56\}$.

Grain v1

Tương tự như Grain v0, Grain v1 cũng sử dụng $k = 80$ và số bits của đầu ra là $l = 64$. Tuy nhiên các bit đầu ra cua Grain v1 được định nghĩa khác với Grain v0: ${z^1}_i = \sum_{i\in A_1} b_{k+i} + \mathrm{h}_1(s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}, b_{i+63})$

Trong đó $A_1 = \{1, 2, 4, 10, 31, 43, 56\}$.

Grain-128

Thuật toán Grain-128 có đầu vào $k = 128$ và đầu ra $l = 96$. Hàm của LFSR được định nghĩa như sau: $\mathrm{f}_{128}(\mathrm{x}) = 1 + x^{32} + x^{47} + x^{58} + x^{90} + x^{121} + x^{128}$. Hàm của NFSR được định nghĩa như sau: $\mathrm{g}_{128}(\mathrm{x}) = 1 + x^{32} + x^{37} + x^{72} + x^{102} + x^{128} + x^{44}x^{60} + x^{61}x^{125} + x^{63}x^{67} + x^{63}x^{67} + x^{69}x^{101} + x^{88}x^{80} + x^{110}x^{111} + x^{115}x^{117}$

Bộ lọc: $\mathrm{h}_{128}(\mathrm{x}) = x_0x_1 + x_2x_3 + x_4x_5 + x_6x_7 + x_0x_4x_8$

Đầu ra:

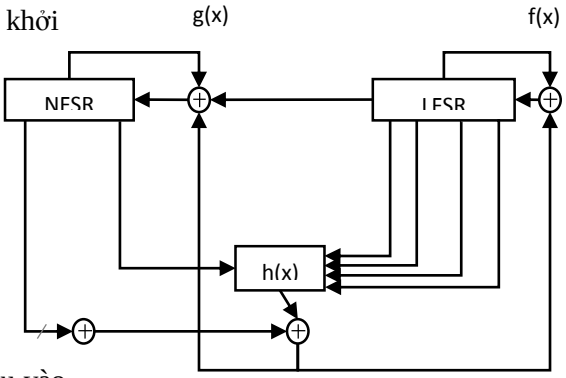
$${z^{128}}_i = \sum_{i\in A_{128}} b_{k+i} + s_{93+i} + \mathrm{h}_{128}(b_{i+12}, s_{i+8}, s_{i+13}, s_{i+20}, b_{i+95}, s_{i+42}, s_{i+60}, s_{i+79}, s_{i+95})$$

Trong đó $A_{128} = \{2, 15, 36, 45, 64, 72, 89\}$.

Tạo khóa:

Tạo khóa

Trước khi tạo ra bất kỳ khóa dòng nào, hệ mã hóa cần khởi tạo khóa và giá trị IV. Khóa sẽ có k bits $k_i, 0 \leq i \leq 79$ và các bit của giá trị IV được xác định bởi $IV_i, 0 \leq i \leq 63$.



Để khởi tạo khóa, đầu tiên ta sử dụng NFSR với khóa $b_i = k_i, 0 \leq i \leq 79$, sử dụng 64 bits đầu tiên của LFSR với giá trị IV là $s_i = IV_i, 0 \leq i \leq 63$. Các bits còn lại của LFSR được xác định bởi $s_i = 1_i, 64 \leq i \leq 79$. Tiếp theo, thuật toán mã hóa được thực hiện 160 lần nhưng không sinh đầu ra trong bất kỳ lần chạy nào, thay vào đó hàm đầu ra sẽ đưa kết quả trở lại và XOR với đầu vào của cả LFSR và NFSR.

3.3 So sánh với các hệ mã hóa nhẹ khác

Thuật toán này cho phép thực hiện song song 16 mã hóa khác nhau, triển khai nhanh hơn, với chi phí sử dụng ít hơn và đem lại hiệu quả cao hơn. Tính hiệu quả của phần cứng là tỷ lệ thông lượng với điện tích sử dụng trong thuật toán, thuật toán Grain có tính hiệu quả phần cứng cao hơn Trivium (77.28 > 38.48).

Mã pháp	Số bit khóa	Số bit khối	Chu kỳ xung nhịp trên một khối	Thông lượng ở 100 MHz (Kbps)	Xử lý logic	Diện tích (GEs)
Mã khối						
Present	80	64	32	200	0,18 µm	1.570
Hight	128	64	34	188	0,25 µm	3.048
mCrypton	96	64	13	492	0,13 µm	2.681
Các mã dòng						
Trivium	80	1	1	100	0,13 µm	2.599
Grain	80	1	1	100	0,13 µm	1.294

Các cuộc tấn công vào hệ mã này để tìm kiếm chìa khóa đầy đủ cần có yêu cầu phức tạp tính toán không thấp hơn 2^{80}. Trong phiên bản gốc v0, tác giả khẳng định: "Grain cung cấp một bảo mật cao hơn so với một số thuật toán mã hóa cũng được biết đến khác, dự định sẽ được sử dụng trong các ứng dụng phần cứng. Ví dụ như trong mã hóa của E0 được sử dụng trong Bluetoot và A5/1 sử dụng trog GSM. So với E0 và A5/1, Grain cung cấp sự bảo mật cao hơn trong khi yêu cầu một phần cứng nhỏ hơn".

3.4 Điểm yếu

Phương pháp tấn công tính toán giá trị Key-IV yếu

Một điểm yếu của Grain chính là Key – IV. Trình tự một keystream tạo ra bởi NFSR rất dễ bị tấn công qua các phương pháp thông dụng như xấp xỉ tuyến tính, chu kỳ ngắn. Trong thực tế, sau 2k lần chạy, trạng thái của LFSR có thể trở về 0. Với phương pháp này Walsh tìm ra 264/264/296 key – IV yếu trong tổng số 2144/2144/2224 key – IV và để tìm ra được các key – IV yếu cần 212.6/244.2/286 bit khóa dòng và 215.8/247.5/2104.2 phép tính cho mỗi phiên bản Grain.

Version	Grain v0	Grain v1
Key	0x6f22a2a70e1c363b62af	0xf57e358ecae6b3dc683d
IV	0x44b604a4d4479eb4	0x97652a7f1a112415
B_{160}	0xc2ced7db3189a9ad94b8	0xd99ea5abb8d0129212c7
S_{160}	0x00000000000000000000	0x00000000000000000000
Version	Grain-128	
Key	0xfd6af0ff0ad9bdad7037b91ef1b9cc13	
IV	0x014d3e274f8d3528ddad4310	
B_{160}	0xc1bc1c087a79b533f9018d230df2e744	
S_{160}	0x00000000000000000000000000000000	

Hình 5: Điểm yếu của giá trị IV trong Grain

Phương pháp tấn công khôi phục Key-IV

Phương pháp tấn công khôi phục Key – IV được Grobner, XL Zhuang-Zi sử dụng để giải quyết bài toán NP-khó trong quá trình tìm Key-IV qua phân tích đại số. Với phương pháp này, hai nhà khoa học đã có thể khôi phục các khóa bí mật 150 bits trong khoảng 2 giây cho Grain v0, Grain v1 và tìm ra chìa khóa của Grain-128 với khoảng 100 bits sau 293.8 phép tính.

Một số phương pháp tấn công khác

Ngoài những phương pháp trên, việc tấn công vào hệ mật Grain còn là niềm đam mê của nhiều nhà nghiên cứu. Với phương pháp do Itai Dinur and Adi Shamir đề xuất để phá vỡ cấu trúc của Grain-128: Dynamic Cube Attacks [12] tìm ra khóa bí mật bằng cách khai thác các kết quả thu được từ cube tester

có thể khôi phục toàn bộ 128 bits của Grain khi số lượng vòng khởi tạo của Grain-128 giảm xuống 207. Hay với phương pháp tấn công bằng đại số điển hình vào mật mã dòng, kẻ thám mã có thể dò ra được đầu ra của hàm NFSR và LFSR. Hay cuộc tấn công Time/Memory/Data Tradeoff có thể phá mã Grain với độ phức tạp tính toán là O(280)...

4. MỘT SỐ ỨNG DỤNG TRONG IoT

Người ta ước tính đến năm 2020 sẽ có hơn 50 tỷ thiết bị kết nối internet, nghĩa là mỗi người trên trái đất trung bình sẽ có 6,6 đồ vật trực tuyến. Trái đất sẽ được che phủ bởi hàng triệu cảm biến thu thập thông tin và tải lên internet. Các ngôi nhà thông minh sẽ được xây dựng, trong các ngôi nhà đó các thiết bị sẽ được kết nối, ví dụ như, ổ khóa thông minh, tủ lạnh thông minh, tivi thông minh, …

Đó chỉ là một số ứng dụng của IoT. Ngoài ra IoT được ứng dụng trong y tế, trong khai thác mỏ an toàn và dự đoán thiên tai được chính xác hơn. Với rất nhiều ứng dụng của IoT nhằm đóng góp vào sự phát triển kinh tế, chăm sóc sức khỏe, giao thông vận tải và đời sống tốt hơn cho công chúng , thì IoT phải cung cấp điều kiện đầy đủ cho việc bảo mật dữ liệu. Đây chính là mảnh đất ứng dụng của các hệ mật mã nhẹ. Các hệ mật mã nhẹ phù hợp với các thiết bị trong IoT, các thiết bị với tài nguyên hạn chế.

Công nghệ RFID

Công nghệ RFID (Radio Frequency Identification, nhận dạng bằng sóng vô tuyến) được tin là công nghệ cho phép kết nối vạn vật.

RFID là một phương pháp nhận dạng tự động dựa trên việc lưu trữ dữ liệu từ xa, sử dụng thiết bị thẻ RFID và một đầu đọc RFID. Một hệ thống RFID tối thiểu hồm những thiết bị sau:

1. Thẻ RFID (RFID Tag, còn được gọi là transponder): là một thẻ gắn chíp + Anten
   Có 02 loại: RFID passive tag và active tag:
   - Passive tags: Không cần nguồn ngoài và nhận nằng lượng từ thiết bị đọc. Khoảng cách đọc ngắn.
   - Active tags: Được nuôi bằng PIN, sử dụng với khoảng cách đọc lớn
2. Reader hoặc sensor (cái cảm biến): để đọc thông tin từ các thẻ, có thể đặt cố định hoặc lưu động.
3. Antenna: là thiết bị liên kết giữa thẻ và thiết bị đọc. Thiết bị đọc phát xạ tín hiệu sóng để kích hoạt và truyền nhận với thẻ.
4. Server: nhu nhận, xử lý dữ liệu, phục vụ giám sát, thống kê, điều khiển,..

Điểm nổi bật của RFID là công nghệ không sử dụng tia sáng như mã vạch, không tiếp xúc trực tiếp. Một vài loại thẻ có thể được đọc xuyên qua các môi trường, vật liệu như Bê tông, tuyết, sương mù, băng đá, sơn, và các điều kiện môi trường thách thức khác mà mã vạch và các công nghệ khác không thể phát huy hiệu quả.

Thẻ RFID có thể đọc trong khoảng thời gian < 10ms.

Thẻ RFID được đưa vào sử dụng trong rất nhiều lĩnh vực như: Quản lý nhân sự, quản lý hàng hóa vào/ra siêu thị, nhà kho, ... theo dõi động vật, quản lý xe cộ qua trạm thu phí, làm thẻ hộ chiếu …

RFID là công nghệ hiện đại giúp nông dân tăng năng suất, giảm chi phí đầu tư

Khi được gắn lên nông sản, thẻ RFID cung cấp thông tin giúp kiểm soát theo quá trình, từ sản xuất, đóng gói, bảo quản, đến vận chuyển,… Nhờ đó người nông dân vừa tăng năng suất chất lượng cho sản phẩm đầu ra, vừa tạo dựng được niềm tin với người mua.

Đối với gia súc được gắn thẻ RFID, người nông dân sẽ xác định vị trí, nguồn gốc, các chỉ số sinh lý… Từ đó có điều chỉnh chế độ ăn uống thích hợp cho gia súc, mặt khác có thể nhanh chóng kiểm soát khi dịch bệnh bùng phát

Công nghệ RFID giúp kiểm soát phương tiện vận chuyển

Thiết bị ghi đọc có thể được bố trí tại các trạm xăng, cổng cảng hoặc các điểm vào cảng khác nhằm cho phép các phương tiện ra vào cảng đồng thời lưu trữ lại các thông tin về thời điểm thực tế mà xe vận

chuyển hoặc container vào hoặc ra bãi cảng. Ngoài ra, thẻ nhận dạng nhân viên có thể được sử dụng để kiểm soát xem có đúng tài xế đi đúng xe vận chuyển và xếp đúng đơn vị hàng hay không?

Ứng dụng công nghệ RFID với hệ thống chuông gọi phục vụ không dây đang được triển khai rộng rãi với mức độ tiện dụng và chi phí thấp

Hệ thống bao gồm 2 thành phần chính:

- Bộ phát tín hiệu (các nút chuông, trung tâm gọi số): Các nút chuông không dây được đặt tại các bàn, phòng hoặc giường (tùy theo không gian của bạn).
- Bộ nhận tín hiệu (Bảng hiển thị, đồng hồ báo tin, bộ đàm, pager): được lắp đặt ở quầy phục vụ/phòng trực, đeo trực tiếp trên tay hoặc gắn ở áo (đối với đồng hồ báo tin)

Kèm một số thiết bị và phần mềm liên quan như: repeater, máy tính, phần mềm lấy dữ liệu thông tin (số lần gọi, số bàn/phòng gọi, thời gian gọi…). Phần này chỉ sử dụng khi thực sự cần thiết như theo dõi trong bệnh viện, nhà xưởng theo một mục đích nào đó. Cách hoạt động rất đơn giản: Khi khách hàng cần gọi phục vụ, chỉ cần nhấn nút chuông, số phòng sẽ hiển thị lên trên các bộ nhận tín hiệu. Từ đó phục vụ sẽ biết được nơi nào đang cần gọi mình.

TÀI LIỆU THAM KHẢO

[1]. G. Leander and A. Poschmann, *In Arithmetic of Finite Fields, First International Workshop - WAIFI 2007, volume 4547 of Lecture Notes in Computer Science*, pages 159 - 176, Springer 2007.

[2]. K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai. Piccolo, *Cryptographic Hardware and Embedded Systems – CHES 2011, volume 6917 of Lecture Notes in Computer Science*, pages 342 - 357, Springer 2011.

[3]. Miroslav Kneˇzevi´c, Ventzislav Nikov, and Peter Rombouts, *Low Latency Encryption - Is "Lightweight = Light + Wait"*, NXP Semiconductors, Leuven, Belgium 2015.

[4]. Miroslav Kneˇzevi´c, *Lightweight Cryptography: from Smallest to Fastest*, NXP Semiconductors, July 2015.

[5]. Nicky Mouha, *The Design Space of Lightweight Cryptography*, Dept. Electrical Engineering-ESAT/COSIC, KU Leuven, Leuven and iMinds, Ghent, Belgium

[6]. Muhammad Usman , Irfan Ahmed , M. Imran Aslam , Shujaat Khan and Usman Ali Shah, SIT, *A Lightweight Encryption Algorithm for Secure Internet of Things*, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 1, 2017.

[7]. M. Hell, T. Johansson, A. Maximov, and W. Meier, *The Grain Family of Stream Ciphers*, In M. Robshaw and O. Billet Editors, New Stream Cipher Designs, LNCS 4986, pp. 179-190, 2008.

[8]. M. Hell, T. Jonasson, and W. Meier. Grain, *A Stream Cipher for Constrained Enviroments*, ECRYPT Stream Cipher Project Report 2005/001, 2005. Available at http://www.ecrypt.eu.org/stream.

[9]. Yi Lu, http://lasecwww.epfl.ch/~vaudenay/ (2004), *Cryptanalysis of Bluetooth Keystream Generator Two-Level E0 (PDF)*, Advances in Cryptology - Asiacrypt 2004, LNCS vol. 3329, pp.483-499, Springer, 2004.

[10]. Côme Berbain, Henri Gilbert, Alexander Maximov (2006-01-02), *Cryptanalysis of Grain* (PDF).

[11]. Haina Zhang, Xiaoyun Wang, *Cryptanalysic of Stream Cipher Graim Family*, https://eprint.iacr.org, 2009.

[12]. Itai Dinur and Adi Shamir - Computer Science department The Weizmann Institute Rehovot 76100, Israel, *Breaking Grain-128 with Dynamic Cube Attacks*, International Association for Cryptologic Research ,2011.

[13]. Alfred J. Menezes, Paul C. Van Oorchot, Scott A. Vanstone, *Hanbook of Applied Cryptography*, CRC Press: Boca Raton – New York – London – Tokyo, 2000.

ABSTRACT

RESEARCH OF SOME LIGHTWEIGHT AND APPLY IN IoT

It is estimated that by 2020 there will be more than 50 billion internet connected devices, meaning that each person on Earth will have an average of 6.6 online items. The Earth will be covered by millions of sensors to crawling and uploading to the internet. To ensure secure connections, these devices need to have the necessary security, low power consumption, memory, and logic ports. These are lightweight cryptographic systems, including block lightweight, stream lightweight, and authentication code lightweight. In this report we introduce some cryptosystems in lightweight cryptography, outlining their strengths and weaknesses. The lightweight systems is research by Klein, Led, Present, Mini - AES, Mcrypyon and Katan. The algorithm we introduced is Grain. The results can be used as reference material for lightweight and IoT coders.

Keywords: Lightweight, Block Cipher, Stream Cipher, IoT, Present, Grain, Delay, Performance, Safety, ...

Received date, 7th April, 2017
Revised manuscript, 10th May, 2017
Published, 09th June, 2017

Author affiliations:
[1] Nhóm Nghiên cứu KH & CN Mật Mã UET-CRYPT, Trường Đại học Công nghệ - ĐHQG HN