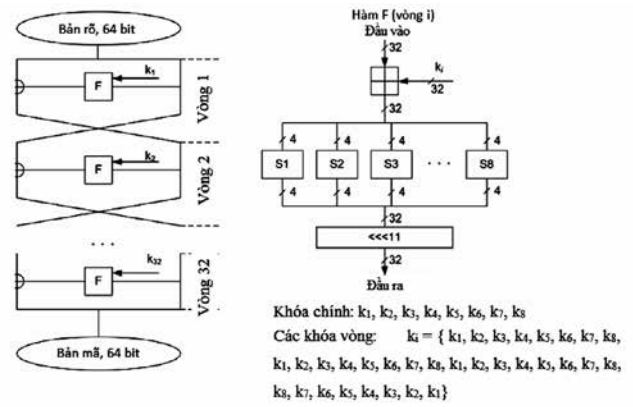
GIỚI THIỆU VỀ THUẬT TOÁN MÃ HÓA MAGMA CỦA LIÊN BANG NGA

Trần Hồng Thái
Viện Khoa học Công nghệ mật mã, Ban Cơ yếu Chính phủ

Bảo mật và an toàn thông tin đóng vai trò then chốt, là yếu tố tiên quyết để triển khai ứng dụng công nghệ thông tin, giao dịch điện tử cho lĩnh vực kinh tế - xã hội. Trong bảo mật và an toàn thông tin, kỹ thuật mật mã đóng vai trò đặc biệt quan trọng, do đó, việc chuẩn hóa các thuật toán mật mã sử dụng cho lĩnh vực kinh tế - xã hội luôn được các quốc gia trên thế giới quan tâm, cập nhật và bổ sung. Bài báo báo này tổng hợp ngắn gọn về nguyên lý thiết kế và độ an toàn kháng lại các tấn công thám mã của thuật toán mã hóa Magma.

Theo tiêu chuẩn GOST R 34.12-2015, thuật toán mã khối GOST 28147-89 được đặt tên là chuẩn mã khối Magma. Đây là thuật toán được phát triển từ những năm 1970 và phân loại "tối mật", sau đó hạ xuống "mật" vào năm 1990. Đến năm 1994, thuật toán này đã được công bố công khai. Magma là thuật toán mã khối đối xứng theo kiến trúc mạng Feistel đơn giản, xử lý với các khối rõ/mã có kích cỡ 64 bit và khóa có kích cỡ 256 bit. Với chi phí cài đặt khá thấp, Magma được cài đặt và sử dụng phổ biến. Nó sử dụng các thành phần và cấu trúc đơn giản như các S-hộp với kích cỡ đầu vào/ra 4 bit, các phép cộng module 2^{32}, phép dịch vòng trái và lược đồ khóa rất đơn giản. Khối dữ liệu được chia thành 2 nửa khối và được biến đổi qua 32 vòng mã hóa (hàm vòng). Trong mỗi vòng, vế phải của các thông điệp bản rõ được xử lý qua hàm F (hàm vòng). Dữ liệu được biến đổi bởi ba phép tính mật mã: cộng hoặc trừ theo modulo 2^{32} giữa dữ liệu với khóa con, thay thế dữ liệu bởi các S-hộp và dịch vòng sang trái 11 vị trí. Đầu ra của hàm F được cộng modulo 2 (cộng XOR) với nửa bên trái của bản rõ, sau đó đổi chỗ hai bên trái và phải ở vòng tiếp theo. Ở vòng cuối cùng, thuật toán không đổi chỗ hai nửa bên trái và bên phải. Sơ đồ mã hóa tổng thể của thuật toán Magma được mô tả trong Hình 1.



Hình 1: Sơ đồ mã hóa tổng thể của thuật toán mã hóa Magma

Magma sử dụng 8 S-hộp 4×4, nghĩa là chuyển 4 bit đầu vào thành 4 bit đầu ra. Khóa bí mật gồm 256 bit và được biểu diễn như một dãy 8 từ (word) 32 bit: k_1, k_2, k_3, k_4, k_5, k_6, k_7 và k_8. Trong mỗi vòng mã hóa, một trong các word 32 bit được sử dụng như một khóa con. Nguyên tắc tính khóa con vòng như sau: từ vòng 1 tới vòng 24 thứ tự là thẳng, nghĩa là được sử dụng một cách lặp lại (k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8), từ vòng 25 tới vòng 32 thứ tự được đảo ngược (k_8, k_7, k_6, k_5, k_4, k_3, k_2, k_1).

Một số đánh giá độ an toàn về Magma

Từ khi được công bố công khai năm 1994, Magma được cộng đồng mật mã trên thế giới quan tâm, phân tích và đánh giá. Dưới đây là tổng hợp một số kết quả chính trong việc phân tích, thám mã lên thuật toán gốc GOST 28147-89.

Một số kết quả về thám mã vi sai

Cho tới nay, có một số kết quả nổi bật về việc áp dụng thám mã vi sai lên thuật toán GOST 28147-89 như sau:

Kết quả của Seki và Kaneko [4]:

Nhóm tác giả chỉ ra khả năng tấn công GOST 28147-89 với số vòng rút gọn sử dụng tập của các đặc trưng vi sai. Trong trường hợp sử dụng bộ S-hộp đã xác định cho Ngân hàng Trung ương Liên bang Nga, khi đó trung bình cần 2^{51} bản rõ lựa chọn thì nhận được khóa của 13 vòng GOST 28147-89. Trường hợp mà các khóa tạo ra xác suất đặc trưng vi sai là cao nhất, thì GOST 28147-89 17 vòng có thể bị tấn công. Với 2^{56} bản rõ lựa chọn, ta có thể khôi phục được khóa của GOST 28147-89 21 vòng.

Các tác giả cũng chỉ ra tấn công này có thể áp dụng được với các S-hộp được sinh ngẫu nhiên. Trung bình 12 vòng của GOST có thể bị tấn công với một tập của các đặc trưng vi sai và 19 vòng của GOST có thể bị tấn công khi kết hợp với tấn công khóa quan hệ.

Kết quả của Babenko và cộng sự [1]:

Babenko và cộng sự thực hiện phân tích tìm kiếm đặc trưng vi sai có xác suất cao cho mã pháp GOST 28147-89. Nhóm tác giả đưa ra chiến lược tìm kiếm bằng cách hạn chế không gian tìm kiếm. Một khẳng định quan trọng đóng vai trò cơ sở cho tấn công là sai khác qua phép cộng modulo 2^{32} sẽ không thay đổi với xác suất ½.

Tấn công vi sai cải tiến của Courtois và Misztal [2]:

Nhóm Courtois và cộng sự nghiên cứu đưa ra một khái niệm rộng hơn là *vi sai gộp* (Aggregated Differential) là tập các đặc trưng vi sai có thể mà có sai khác đầu vào *a* bất kỳ và sai khác đầu ra *b* bất kỳ. Xác suất gắn với nó là xác suất cao nhất của các đặc trưng trong tập đó (thỏa mãn theo các mẫu sai khác đầu vào và sai khác đầu ra cụ thể).

Khi đó, việc tìm kiếm các đặc trưng vi sai được hạn chế với việc tìm xác suất lớn nhất cho vi sai gộp. Courtois hạn chế với vi sai gộp dạng (Δ, Δ) là tập sai khác lên 64 bit có từ 1 - 14 bit tích cực theo mẫu mặt che. Có tổng số 2^{14} - 1 sai khác trong tập này. Các đặc trưng vi sai mà Courtois và Misztal tìm được có xác suất cao hơn so với kết quả trước đó. Thêm vào đó, các tác giả đề xuất sử dụng tấn công vi sai bội (nghĩa là sử dụng nhiều vi sai mà có cùng mẫu sai khác trong một vi sai gộp) lên GOST 28147-89. Như vậy, khả năng tấn công có thể sẽ hiệu quả hơn.

Tuy nhiên, tấn công này chưa được cộng đồng thế giới đánh giá cao về tính chính xác và thực tế. Bởi với một tập vi sai cụ thể (ví dụ: 0x70707070, 0x07070707) theo đánh giá lý thuyết có giá trị xác suất là 2^{-160} qua 32 vòng mã, nhưng theo đánh giá của Courtois thì giá trị này chỉ là 2^{-40}. Dẫn tới, có thể có rất nhiều dự tuyển nhầm (false positives). Trong thực tế, tấn công sẽ không thể xử lý với các trường hợp này.

Một số kết quả thám mã vi sai khóa có quan hệ

Các tấn công khóa có quan hệ là mô hình tấn công không thực tế, vì chúng đòi hỏi phải nắm được quan hệ giữa các cặp khóa được sử dụng. Tuy nhiên, theo giả thiết này, có một số kết quả phân tích đáng chú ý bao gồm:

Tấn công khóa quan hệ của Seki và Kaneko [4]:

Seki và Kaneko đưa ra một thám mã vi sai cho trường hợp khóa có quan hệ và chỉ ra đặc trưng cụ thể. Các tác giả chỉ ra rằng, nếu sử dụng các S-hộp yếu thì phương pháp này có thể tấn công tới 27 vòng.

Tấn công vi sai khóa quan hệ của Ko [5]:

Ko và cộng sự đưa ra một tấn công cải tiến vi sai khóa quan hệ khác lên GOST 28147-89 dựa trên kết quả của Seki. Trước tiên, Ko xây dựng một đặc trưng vi sai khóa quan hệ 6 vòng khác từ vòng 25 qua vòng 30. Sau đó, kết hợp với đặc trưng vi sai khóa quan hệ 24 vòng của Seki và Kaneko để tạo thành một đặc trưng vi sai khóa quan hệ 30 vòng của GOST 28147-89 với xác suất khoảng $2^{-23,33}$. Theo đó, nếu chọn 2^{35} cặp văn bản thì xác suất tấn công thành công khoảng 0,917 với độ phức tạp cỡ 2×2^{35} phép mã.

Tấn công vi sai khóa quan hệ của M. Pudovkina, G. Khoruzhenko [7]:

Nhóm tác giả M. Pudovkina và G. Khoruzhenko đưa ra tấn công khôi phục khóa với S-hộp tùy ý sử dụng 12 khóa có quan hệ để tìm ra toàn bộ 256 bit khóa của GOST 28147-89. Phương pháp được đề xuất gồm 2 bước chính.

Bước đầu tiên, xác định các khóa ứng cử viên của vòng 31-32, áp dụng tấn công Boomerang khóa có quan hệ sử dụng 4 khóa có quan hệ.

Bước thứ hai, tìm và xác định các khóa con từ vòng 25-30 bằng cách sử dụng tấn công Boomerang khóa có quan hệ với các khóa có quan hệ.

Các tác giả ước lượng rằng phương pháp này cần khoảng $2^{42,9}$ cặp bản rõ chọn lọc, độ phức tạp cỡ $2^{61,39}$ và xác suất thành công là 0,992.

Các tấn công khai thác tính chất phản xạ và điểm bất động

Đây là các tấn công khai thác các tính chất do cấu trúc đặc thù của GOST 28147-89. Tính chất phản xạ của GOST được Kara [8] chỉ ra vào năm 2007 và dẫn tới một tấn công lên GOST đủ vòng. Tính chất này được nghiên cứu chuyên sâu bởi Courtois [3], Dinur [9] và Isobe [10]. Isobe thiết lập một tấn công sử dụng khóa bất kỳ với độ phức tạp thời gian lên tới 2^{225}. Còn Dinur và Dunkelman cải tiến tấn công này, giảm độ phức tạp thời gian xuống còn 2^{192} sử dụng toàn bộ không gian bản rõ (2^{64} khối rõ có thể). Dinur giới thiệu một phiên bản mới của phương pháp gặp ở giữa, gọi là gặp ở giữa 2 chiều (2DMITM).

Tấn công của họ dựa trên việc áp dụng phương pháp 2DMITM trên 8 vòng của GOST 2^{32} lần. Sau đó Kara và Ferhat Karakoç đưa một tấn công cải tiến, áp dụng 2DMITM với độ phức tạp thời gian là 2^{129} và sử dụng 2^{32} bản rõ/mã được lựa chọn; thay vì sử dụng toàn bộ không gian bản rõ. Với ưu điểm chính là chỉ áp dụng tấn công 2DMITM trên

8 vòng của GOST hai lần. Thêm vào đó, tấn công này sử dụng một tập khóa yếu gồm 2^{192} khóa, điều này chỉ ra rằng, độ an toàn của thuật toán GOST chỉ tương đương với 192 bit tương ứng với các khóa yếu dạng này.

Tấn công của Isobe [10]

Isobe là người đầu tiên đưa ra phương pháp thám mã lên GOST mà không sử dụng các lớp khóa yếu (chỉ dùng 1 khóa mã). *Tấn công gặp ở giữa cơ sở lên 8* vòng *GOST:*

Giả sử ta có hai cặp đầu vào/ra 8 vòng của GOST là (I,O) và (I^*,O^*). Tấn công MITM cơ sở lên 8 vòng GOST sử dụng hai cặp này như sau:

Bước 1: Với mỗi giá trị có thể của khóa k_1-k_4 (có tất cả 2^{128} giá trị) tính:

$$U = F_K[1,4](I)$$

$$U^* = F_K[1,4](I^*)$$

Lưu U và U^* vào một danh sách, sắp xếp chúng tương ứng với 128 bit khóa k_1-k_4 đã sử dụng. Gọi mỗi phần tử của danh sách này là $(U,U^*)_{k_1-k_4}$. Số phần tử của danh sách này là 2^{128} phần tử.

Bước 2: Với mỗi giá trị có thể của k_5-k_8, tính

$$V = F_K{}^{-1}[5,8](O)$$

$$V^* = F_K{}^{-1}[5,8](O^*)$$

Ở đây, phép $F_k{}^{-1}[5,8](O)$ là phép giải mã O, $F_k{}^{-1}[5,8](O^*)$ là phép giải mã O^*. Lưu V và V^* vào một danh sách, sắp xếp chúng tương ứng với 128 bit khóa k_5-k_8 đã sử dụng. Gọi mỗi phần tử của danh sách này là $(V,V^*)_{k_5-k_8}$. Số phần tử của danh sách này là 2^{128} phần tử.

Bước 3: So sánh (so khớp) các giá trị $(U,U^*)_{k_1-k_4}$ thu được ở bước 1 và $(V,V^*)_{k_5-k_8}$ thu được tại bước 2. Nếu $(U,U^*)_{k_1-k_4} = (V,V^*)_{k_5-k_8}$ thu được một đề xuất cho khóa đầy đủ 256 bit $k_1 - k_8$ ứng với hai giá trị này.

Khôi phục lại khóa chính xác bằng cách sử dụng thêm một số cặp rõ/mã 32 vòng đã biết.

Độ phức tạp *về thời gian:* Tổng độ phức tạp thời gian của tấn công cỡ $O(2^{128})$.

Độ phức tạp về *bộ nhớ:* độ phức tạp về bộ nhớ của tấn công là khoảng 2^{128} từ (word) 256 bit, dùng để lưu 2^{128} giá trị $(U,U^*)_{k_1-k_4}$ trong bước 1, 2^{128} giá trị $(V,V^*)_{k_5-k_8}$ trong bước 2.

Tấn công phản xạ - gặp ở giữa (R-MITM):

Dựa trên ý tưởng về tấn công phản xạ của Kara, Isobe đưa một tấn công kết hợp giữa phản xạ và gặp ở giữa (tấn công R-MITM). Ở đây, ta giả sử áp dụng tấn công phản xạ để đoán được bản mã trung gian ở giữa 8 vòng, thì ta có thể áp dụng tấn công gặp nhau ở giữa với 4 vòng phía trước hoặc phía sau.

Giả sử ta có cặp đầu vào và đầu ra của 4 vòng, với mỗi lần đoán giá trị của khóa $k_1 - k_2$, ta có thể tính các giá trị trung gian sau vòng thứ 2 và tìm các khóa $k_3 - k_4$ mà thỏa mãn cặp đầu vào/ra ở trên.

Trong [11], Dinur đã gọi các khóa này là các khóa tương đương. Có tất cả 2^{64} giá trị khóa có thể $k_1 - k_2$ mà tạo ra cặp đầu vào/ra 4 vòng ở trên.

Tấn công của Dinur và Dunkelman [9]:

Dinur và Dunkelman cải tiến tiếp các tấn công của Isobe để làm giảm độ phức tạp của tấn công và đưa ra một số phương pháp tấn công khác nhau lên GOST 28147-89. Tuy nhiên, tấn công này phải sử dụng lớp khóa yếu của GOST 28147-89 (vì cần tới các điểm bất động). Nhưng tấn công này vẫn được xem là tấn công tốt nhất lên thuật toán GOST 28147-89. Đó là tấn công gặp nhau ở giữa cải tiến cho 8 vòng và tấn công 2 chiều cải tiến. Ưu điểm của tấn công MITM cải tiến mà Dinur đề xuất là làm việc được với tất cả các phiên bản của GOST. Có thể hình dung rằng tấn công cải tiến của Dinur (là tấn công dựa trên phản xạ, MITM và điểm bất động) suy dẫn tấn công lên GOST 32 vòng về tấn công lên 8 vòng để tìm điểm bất động. Tấn công gặp nhau ở giữa thực chất là việc phân hoạch ra thành 2 tấn công lên 4 vòng (nói đơn giản là chia không gian khóa thành 2 phần bằng nhau và tìm phần giao nhau ở giữa). Hình thức này được gọi là tấn công 4 vòng phần đỉnh và tấn công lên 4 vòng phần đáy - là tấn công gặp nhau ở giữa theo chiều dọc.

Dinur cho rằng có thể tận dụng tính khuếch tán chậm của GOST, vì chỉ sử dụng phép dịch vòng

sang trái 11 vị trí trong hàm vòng. Để làm giảm độ phức tạp lưu trữ trong tấn công MITM, Dinur đề xuất thay vì xét cả khối 32 bit trong mỗi vòng mã hóa, thì sẽ thực hiện tấn công trên theo từng bộ 4 bit (nibble). Khi đó, cần phải xét thêm các bit nhớ và sự lan truyền bit nhớ sang một số bộ nibble khác có thể. Đồng thời, độ phức tạp tính toán tăng thêm, nhưng bù lại độ phức tạp về bộ nhớ lưu trữ lại giảm đi đáng kể. Đây chính là giải pháp cải tiến để làm giảm bộ nhớ của Dinur. Giải pháp cho thấy, các bit khóa ứng cử viên cũng được duyệt theo các nibble có liên quan. Hơn nữa, khi xét 2 nibble liên tiếp nhau, thì có thể tạo ra các tập khóa ứng cử viên (cho từng nibble) giao nhau (vì bị ảnh hưởng bởi phép dịch vòng), giúp ta xác định các bit khóa tốt hơn trong việc tìm khóa. Đây chính là ý tưởng để phát triển tiếp tấn công gặp nhau ở giữa 2 chiều.

Do vậy, Dinur đề xuất giải pháp phân hoạch các tấn công lên 4 vòng phần đỉnh (hoặc phần đáy) trong tấn công ở trên thành 2 nửa theo chiều ngang (2 tấn công song song). Do ảnh hưởng của phép dịch vòng và phép cộng modulo 2^{32}, nên giữa 2 nửa vẫn có sự liên quan (vì có sự ảnh hưởng của các bit nhớ). Và việc xét phần giao nhau giữa các bit khóa bị ảnh hưởng trong 2 tấn công song song này chính là một dạng tấn công gặp nhau ở giữa theo chiều ngang.

Kết luận

Trong khuôn khổ bài báo này, tác giả không thể trình bày rõ ràng từng tấn công. Cho tới nay, nhóm tác giả Courtois được cho là có nhiều kết quả nghiên cứu, đánh giá nhất với thuật toán mã hóa GOST 28147-89. Tuy nhiên, các kết quả của Courtois và cộng sự không được đánh giá cao, vì dựa trên nhiều giả thiết không thực tế và các kết quả dựa nhiều vào lập luận còn chưa rõ ràng. Một kết quả gần đây nhất của Courtois là rút gọn tấn công đại số lên GOST 28147-89. Ý tưởng là sử dụng tính chất tự đồng dạng của GOST để suy dẫn việc phân tích với đầy đủ 32 vòng mã hóa, sử dụng số lượng lớn các cặp bản rõ/mã trở về bài toán chỉ phân tích với 8 vòng mã hóa sử dụng số lượng nhỏ các cặp bản rõ/mã. Tiếp theo đó, viết các F_K như là hệ các đa thức nhiều biến và tìm nghiệm của hệ phương trình bằng phương pháp tuyến tính hóa mở rộng (XSL). Tác giả đưa ra hơn 20 cách tiếp cận khác nhau để đưa ra các suy dẫn khác nhau. Tuy nhiên, các kết quả suy dẫn để tấn công GOST 28147-89 vẫn còn chưa thực tế. Ví dụ: sử dụng 2^{32} khóa đã biết, 2^{64} bản rõ đã biết và độ phức tạp để tấn công một khóa yếu là 2^{193} lần mã hóa.

Có thể khẳng định rằng, tấn công của Dinur và Dunkelman là tấn công tốt nhất hiện nay lên mã khối GOST 28147-89, khai thác triệt để cấu trúc của thuật toán mã hóa này. Tấn công này là sự kế thừa, kết hợp và phát triển mọi kỹ thuật tấn công đã có lên GOST 28147-89. Theo đó, độ phức tạp tốt nhất là với 2^{32} dữ liệu (bản rõ/mã) đã biết, ta có thể giảm độ phức tạp bộ nhớ từ 2^{64} về 2^{36} mà không thay đổi độ phức tạp thời gian 2^{224}; khi biết trước 2^{64} dữ liệu, ta có thể đồng thời giảm độ phức tạp thời gian về 2^{192} và độ phức tạp bộ nhớ về 2^{36}.

Tài liệu tham khảo

1. Babenko, L. and E. Ishchukova. *Differential analysis of GOST encryption algorithm.* in Proceedings of the 3rd international conference on Security of information and networks. 2010. ACM.
2. Courtois, N.T. and M. Misztal, *Aggregated Differentials and Cryptanalysis of PP-1 and GOST.* Periodica Mathematica Hungarica, 2012. 65(2): p. 177-192.
3. Courtois, N.T., *An Improved Differential Attack on Full GOST.* IACR Cryptology ePrint Archive, 2012: p. 138.
4. Seki, H. and T. Kaneko. *Differential cryptanalysis of reduced rounds of GOST.* in *Selected Areas in Cryptography*. 2001. Springer.
5. Ko, Y., et al. *Related key differential attacks on 27 rounds of XTEA and full-round GOST.* in *Fast Software Encryption*. 2004. Springer.
6. Пудовкина, М.А. and Г.И. Хоруженко, Атаки на алгоритм блочного шифрования ГОСТ 28147-89 с двумя и четырьмя связанными ключами. МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ, 2010: p. 29.
7. Pudovkina, M.A. and G.I. Khoruzhenko, *An attack on the GOST 28147-89 block cipher with 12 related keys.* Matematicheskie Voprosy Kriptografii [Mathematical Aspects of Cryptography], 2013. 4(2): p. 127-152.
8. Kara, O., *Reflection Attacks on Product Ciphers.* IACR Cryptology ePrint Archive, 2007: p. 43.11
9. Dinur, I., O. Dunkelman, and A. Shamir. *Improved attacks on full GOST.* in *Fast Software Encryption*. 2012. Springer.
10. Isobe, T., *A single-key attack on the full GOST block cipher.* Journal of cryptology, 2013. 26(1): p. 172-189.
11. Rudskoy, V. and A. Dmukh. *Algebraic and Differential cryptanalysis of GOST: Fact or Fiction.* in CTCrypt 2012, Workshop on Current Trends in Cryptology, affiliated with 7th International Computer Science Symposium in Russia (CSR-2012). 2012.