ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC BÁCH KHOA

TÓM TẮT BÁO CÁO TỔNG KẾT

ĐỀ TÀI KHOA HỌC VÀ CÔNG NGHỆ CẤP ĐẠI HỌC ĐÀ NẴNG

PHÂN TÍCH ĐỊNH LƯỢNG LUỒNG THÔNG TIN TRONG BẢO MẬT PHẦN MỀM

Mã số: B2016-ĐN02-13

Chủ nhiệm đề tài: TS. NGÔ MINH TRÍ

ĐÀ NẴNG, 05/2018

NHỮNG THÀNH VIÊN THAM GIA NGHIÊN CỨU ĐỀ TÀI			
TT	Họ và tên	Đơn vị công tác và lĩnh vực chuyên môn	Nội dung nghiên cứu cụ thể được giao
1	TS. Ngô Minh Trí	Khoa Điện tử - Viễn thông	Chủ nhiệm
2	TS. Huỳnh Việt Thắng	Khoa Điện tử - Viễn thông	Thành viên chính
3	TS. Nguyễn Quang Như Quỳnh	Khoa Điện tử - Viễn thông	Thành viên
4	KS. Vũ Vân Thanh	Khoa Điện tử - Viễn thông	Thư ký khoa học
ĐƠN VỊ PHỐI HỢP CHÍNH			
Tên đơn vị trong và ngoài nước		Nội dung phối hợp nghiên cứu	Họ và tên người đại diện đơn vị
Trường Đại học Bách khoa Twente, Hà Lan		Tư vấn phần thuật toán	GS Marieke Huisman

MỤC LỤC

ĐẠI HỌC ĐÀ NẴNG

Đơn vị: Trường Đại học Bách Khoa

THÔNG TIN KẾT QUẢ NGHIÊN CỨU

1. Thông tin chung:

- Tên đề tài: PHÂN TÍCH ĐỊNH LƯỢNG LUỒNG THÔNG TIN TRONG BẢO MẬT PHẦN MỀM

- Mã số: B2016-ĐN02-13

- Chủ nhiệm đề tài: TS. Ngô Minh Trí

- Tổ chức chủ trì: Trường Đại học Bách khoa – Đại học Đà Nẵng

- Thời gian thực hiện: 10/2016 – 09/2018

2. Mục tiêu:

- *Mục tiêu 1*: Xây dựng thuật toán phân tích định lượng luồng tin cho các chương trình đa luồng.

- *Mục tiêu 2*: Xây dựng quy trình đảm bảo an toàn, an ninh thông tin cho các ứng dụng, phần mềm.

3. Tính mới và sáng tạo:

Trong thời đại thông tin hiện nay, dữ liệu là một nguồn tài nguyên quý giá. Vì vậy, đảm bảo tính bí mật cho các thông tin quan trọng là một nhiệm vụ có tầm ảnh hưởng đến tất cả các lĩnh vực của cuộc sống. Chính phủ, quân đội, các công ty, các hệ thống tài chính, các dịch vụ trực tuyến đều muốn đảm bảo dữ liệu của mình được an toàn. Nếu những thông tin quan trọng này rơi vào tay kẻ xấu, hậu quả có thể rất nghiêm trọng, nhất là những thông tin liên quan đến an ninh quốc gia. Do đó, chúng ta đang đứng trước thách thức là cần phải làm gì để đảm bảo an toàn cho thông tin, và xác định phương thức nào là hiệu quả nhất cho mục tiêu này. Nhiều phương pháp bảo đảm an toàn, an ninh thông tin cho các ứng dụng đã được đề xuất như mật mã học (cryptography) hay điều khiển truy nhập hệ thống (access control). Tuy các phương pháp này đều hữu dụng nhưng chúng có một giới hạn căn bản: chúng không thể đảm bảo được rằng thông tin trong hệ thống sẽ được bảo mật toàn vẹn từ đầu đến cuối (end-to-end). Một phương pháp mới đang thu hút sự chú ý của cộng đồng bảo mật thông tin gần đây là phương pháp phân tích luồng tin. Phương pháp

phân tích này có thể chỉ ra rằng liệu trong hệ thống có tồn tại sự sao chép từ dữ liệu mật đến dữ liệu công cộng hay không. Khi đó, bất cứ người dùng nào cũng có thể truy xuất được dữ liệu mật thông qua dữ liêu công cộng.

Trong thực tiễn, rất nhiều ứng dụng, như các dịch vụ mạng, các cơ sở dữ liệu hay các hệ điều hành là các chương trình đa luồng trong đó các tác nhiệm được thực thi đồng thời, song song với nhau. Cùng với sự phát triển của các máy tính với các bộ xử lý đa lõi, các chương trình đa luồng đã trở nên rất phổ biến hiện nay. Tuy nhiên, việc đảm bảo an toàn, an ninh thông tin cho các ứng dụng chương trình đa luồng là khá khó khăn. Đó là do chúng ta khó truy vết được sự phụ thuộc lẫn nhau giữa các dòng tin của các tác nhiệm chạy song song với nhau. Hiện tại, tuy có nhiều nhà nghiên cứu quan tâm đến lĩnh vực này nhưng các phương pháp tiếp cận vẫn chưa đạt được hiệu quả mong muốn. Mục tiêu chính của đề tài này là xây dựng một phương pháp hữu hiệu để phân tích tính bảo mật cho các chương trình phần mềm đa luồng.

4. Kết quả nghiên cứu:

Phương pháp phân tích định lượng luồng tin cho các chương trình đa luồng.

5. Sản phẩm:

- Sản phẩm khoa học: 01 bài báo trên tạp chí khoa học chuyên ngành quốc tế (SCI,Q2) và 01 bài báo hội nghị quốc tế (Springer).

(Trong thuyết minh, đề tài chỉ đăng ký bài báo trên tạp chí khoa học chuyên ngành *trong nước*)

- Sản phẩm ứng dụng: Phương pháp/thuật toán phân tích định lượng luồng tin cho các chương trình

- Báo cáo phân tích

- Sản phẩm đào tạo: Thạc sĩ: Dương Tuấn Quang, Đề tài: Phân tích định lượng luồng tin trong bảo mật chương trình đa luồng. Ngành Kỹ thuật Điện tử, K35

(Trong thuyết minh không có sản phẩm đào tạo nhưng trong quá trình thực hiện đề tài, đã đào tạo được 01 ThS.)

6. Phương thức chuyển giao, địa chỉ ứng dụng, tác động và lợi ích mang lại của kết quả nghiên cứu:

- Chuyển giao toàn bộ sản phẩm, gồm: các báo cáo, bài báo khoa học, chương trình minh họa hoạt động của hệ thống.

- Địa chỉ ứng dụng: Khoa Điện tử - Viễn thông, Trường Đại học Bách khoa - Đại học Đà Nẵng.

- Giá trị của đề tài:

+ Về mặt giáo dục – đào tạo: góp phần làm phong phú thêm tài liệu tham khảo cho giảng dạy và nghiên cứu khoa học của giảng viên, sinh viên trong khoa Điện tử - Viễn thông, Công nghệ thông tin, Chương trình tiên tiến, đặc biệt là Chương trình An toàn, an ninh hệ thống thông tin sắp tới của Khoa Công nghệ thông tin.

+ Về mặt khoa học: đóng góp đáng kể của đề tài là đưa ra phương pháp phân tích định lượng luồng tin cho các chương trình. Bên cạnh đó, đề tài cũng cung cấp một giải pháp bảo mật cho các ứng dụng.

Ngày 19 tháng 06 năm 2018

Tổ chức chủ trì	**Chủ nhiệm đề tài**
(ký, họ và tên, đóng dấu)	*(ký, họ và tên)*
PHÓ HIỆU TRƯỞNG	
PGS. TS. LÊ THỊ KIM OANH	Ngô Minh Trí

TỔNG QUAN VỀ ĐỀ TÀI

1. Tổng quan tình hình nghiên cứu thuộc lĩnh vực của đề tài ở trong và ngoài nước

Ngoài nước: các phương pháp phân tính định tính luồng tin ứng dụng trong bảo mật phần mềm đã được nghiên cứu rộng rãi từ nhiều năm nay. Tuy nhiên, các phương pháp định lượng luồng tin bị rò rỉ trong các ứng dụng thì chỉ mới được nghiên cứu gần đây, nhưng chủ yếu chỉ cho các chương trình đơn luồng đơn giản. Cụ thể là qua các công bố sau:

[1] Miguel E. Andrés, Catuscia Palamidessi, Geoffrey Smith: Preface to the special issue on quantitative information flow. Mathematical Structures in Computer Science 25(2): 203-206 (2015)

[2] Catuscia Palamidessi: Quantitative Approaches to the Protection of Private Information: State of the Art and Some Open Challenges. POST 2015: 3-7

[3] Barbara Espinoza, Geoffrey Smith: Min-entropy as a resource. Inf. Comput. 226: 57-75 (2013)

[4] Geoffrey Smith: Recent Developments in Quantitative Information Flow (Invited Tutorial). LICS 2015: 23-31

Trong đề tài này, chúng tôi sẽ nghiên cứu phương pháp định lượng luồng tin cho các chương trình phần mềm đa luồng.

Trong nước: theo như hiểu biết của chủ nhiệm đề tài thì hiện tại, ở Việt Nam không có nhóm nghiên cứu nào đã và đang nghiên cứu lĩnh vực này.

2. Tính cấp thiết của đề tài

Trong thời đại thông tin hiện nay, dữ liệu là một nguồn tài nguyên quý giá. Vì vậy, đảm bảo tính bí mật cho các thông tin quan trọng là một nhiệm vụ có tầm ảnh hưởng đến tất cả các lĩnh vực của cuộc sống. Chính phủ, quân đội, các công ty, các hệ thống tài chính, các dịch vụ trực tuyến đều muốn đảm bảo dữ liệu của mình được an toàn. Nếu những thông tin quan trọng này rơi vào tay kẻ xấu, hậu quả có thể rất nghiêm trọng, nhất là những thông tin liên quan đến an ninh quốc gia. Do đó, chúng ta đang đứng trước thách thức là cần phải làm gì để đảm bảo an toàn cho thông tin, và xác định phương thức nào là hiệu quả nhất cho mục tiêu này. Nhiều phương pháp bảo đảm an toàn, an ninh thông tin cho các ứng dụng đã được đề xuất như mật mã học (cryptography) hay điều khiển truy nhập hệ thống (access control). Tuy các phương pháp này đều hữu dụng nhưng chúng có một giới hạn căn bản: chúng không thể đảm bảo được rằng thông tin trong hệ thống sẽ

được bảo mật toàn vẹn từ đầu đến cuối (end-to-end). Một phương pháp mới đang thu hút sự chú ý của cộng đồng bảo mật thông tin gần đây là phương pháp phân tích luồng tin. Phương pháp phân tích này có thể chỉ ra rằng liệu trong hệ thống có tồn tại sự sao chép từ dữ liệu mật đến dữ liệu công cộng hay không. Khi đó, bất cứ người dùng nào cũng có thể truy xuất được dữ liệu mật thông qua dữ liêu công cộng.

Trong thực tiễn, rất nhiều ứng dụng, như các dịch vụ mạng, các cơ sở dữ liệu hay các hệ điều hành là các chương trình đa luồng trong đó các tác nhiệm được thực thi đồng thời, song song với nhau. Cùng với sự phát triển của các máy tính với các bộ xử lý đa lõi, các chương trình đa luồng đã trở nên rất phổ biến hiện nay. Tuy nhiên, việc đảm bảo an toàn, an ninh thông tin cho các ứng dụng chương trình đa luồng là khá khó khăn. Đó là do chúng ta khó truy vết được sự phụ thuộc lẫn nhau giữa các dòng tin của các tác nhiệm chạy song song với nhau. Hiện tại, tuy có nhiều nhà nghiên cứu quan tâm đến lĩnh vực này nhưng các phương pháp tiếp cận vẫn chưa đạt được hiệu quả mong muốn. Mục tiêu chính của đề tài này là xây dựng một phương pháp hữu hiệu để phân tích tính bảo mật cho các chương trình phần mềm đa luồng.

3. Mục tiêu của đề tài

- *Mục tiêu 1*: Xây dựng thuật toán phân tích định lượng luồng tin cho các chương trình đa luồng.

- *Mục tiêu 2*: Xây dựng quy trình đảm bảo an toàn, an ninh thông tin cho các ứng dụng, phần mềm.

4. Đối tượng, phạm vi nghiên cứu

- *Đối tượng nghiên cứu*:

Nghiên cứu đặc tính các chương trình phần mềm đa luồng.

Nghiên cứu phương pháp phân tích định lượng luồng thông tin cho các chương trình phần mềm đa luồng.

- *Phạm vi nghiên cứu*:

Quy trình bảo mật dựa trên kỹ thuật phân tích định lượng luồng tin cho các chương trình đa luồng.

5. Cách tiếp cận, phương pháp nghiên cứu

- Cách tiếp cận: Kế thừa những công trình nghiên cứu về phân tích định tính luồng tin cho các chương trình đa luồng trước đây của chủ nhiệm đề tài và những nghiên cứu gần đây nhất về phân tích định lượng luồng tin cho các chương trình đơn luồng của các nhóm nghiên cứu khác trên thế

giới, chủ nhiệm đề tài sẽ đề xuất phương pháp và thuật toán để ước tính luồng tin bị rò rỉ trong các ứng dụng đa luồng.

- Phương pháp nghiên cứu:

- Xem xét các công trình liên quan, so sánh và đánh giá các ưu điểm và khuyết điểm của các phương pháp đã có.

- Đề xuất thuật toán phân tích định lượng luồng tin cho các chương trình đa luồng.

- Xây dựng chương trình tính toán

- Kiểm tra tính hiệu quả của phương pháp đề xuất dựa trên việc phân tích và đánh giá các kết quả đạt được so với các phương pháp trước.

6. Nội dung nghiên cứu

Nội dung 1: Tổng quan về các phương pháp phân tích luồng tin

+ Nghiên cứu tổng quan về vấn đề bảo mật thông tin trong các chương trình tính toán

+ Tổng quan về các phương pháp phân tích định tính, định lượng luồng tin

+ Khảo sát các nghiên cứu liên quan trong lĩnh vực của đề tài trong những năm gần đây

Nội dung 2: Entropy và phương pháp tiếp cận phân tích định lượng luồng tin

+ Nghiên cứu các khái niệm entropy được sử dụng trong phân tích định lượng luồng tin

+ Đánh giá tính chính xác của các phương pháp đã có đối với chương trình đa luồng

+ Đề xuất các đại lượng tính toán mới cho các chương trình đa luồng

+ Viết báo cáo khoa học

Nội dung 3: Kỹ thuật/Thuật toán ước lượng luồng tin cho các ứng dụng đa luồng

+ Đề xuất các thuật toán định lượng dòng tin cho các chương trình đa luồng

+ Viết báo cáo khoa học

Nội dung 4: Case studies: Ứng dụng của phương pháp trong các trường hợp thực tiễn

+ Áp dụng các phương pháp vào các chương trình thực tiễn

+ Đánh giá, so sánh kết quả đạt được

+ Viết báo cáo khoa học

Nội dung 5: Kết luận/Đánh giá đề tài/Định hướng phát triển của đề tài

+ Viết báo cáo tổng kết toàn bộ đề tài, trên cơ sở tổng hợp tất cả các công trình nghiên cứu đã công bố liên quan đến đề tài

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT BẢO MẬT THÔNG TIN

1.1. Entropy

Lượng tin riêng của một tin $x \in X$ chỉ có ý nghĩa đối với chính tin đó mà thôi, chứ không phản ánh được giá trị tin tức của nguồn X. Nói cách khác, $I(x)$ chỉ mới đánh giá được về mặt tin tức của một tin khi nó đứng riêng rẽ chứ không đánh giá được tin tức của tập hợp chứa tin đó. Từ đó dẫn đến khái niệm giá trị trung bình củ các thông tin đó. Giá trị trung bình này còn được gọi là lượng tin trung bình như đã trình bày ở trên, hay còn gọi là entropy.

Để tính toán lượng tin trung bình, lý thuyết thông tin sẽ sử dụng khái niệm entropy, $\mathcal{H}$, để xác định sự bất định trong việc dự đoán giá trị của biến ngẫu nhiên. Ở đây, ta sẽ xem xét hai loại entropy và sử dụng các khái niệm này trong khía cạnh bảo mật thông tin sẽ được trình bày ở chương tiếp theo.

1.1.1. Shannon Entropy

Định nghĩa 1.7 [*Shannon entropy [10]*] Gọi $p(x)$ là xác suất của biến X nhận được tại giá trị x, khi đó entropy $\mathcal{H}(X)$ của biến ngẫu nhiên X được định nghĩa như sau:

$$\mathcal{H}(X) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$$

Entropy có đơn vị tính bằng bit, và xác định độ bất định của một biến ngẫu nhiên. Về cơ bản, entropy chính là lượng bit thông tin trung bình dung để mô tả một biến ngẫu nhiên.

Ví dụ 1. Entropy của một biến ngẫu nhiên mô tả quá trình tung hạt xúc xắc (với xác suất xuất hiện của từng mặt là 1/6) là $\sum_{x \in \{1,\dots,6\}} \frac{1}{6} \log_2 \frac{1}{6} \approx 2.58\ bits$. Giả sử hạt xúc xắc có trong lượng không đều khiến cho mặt số sáu luôn luôn xuất hiện trong mỗi lần gieo (hay phân bố xác suất gieo ra mặt số sáu là 1 và xác suất gieo ra các mặt còn lại là 0), entropy của biến ngẫu nhiên lúc này sẽ là $\sum_{x \in \{1,\dots,5\}} 0 \log_2 0 + \sum_{x \in \{6\}} 1 \log_2 1 = 0\ bits$. Điều này có nghĩa là giá trị biến ngẫu nhiên là xác định, luôn dự đoán chính xác được.

Định nghĩa entropy cho một biến ngẫu nhiên có thể mở rộng cho hai biến ngẫu nhiên, gọi là entropy hợp.

Định nghĩa 1.8 [*Entropy liên kết [10]*] Entropy liên kết $\mathcal{H}(X,Y)$ của hai biến ngẫu nhiên rời rạc (X,Y) với xác suất hợp $p(x,y)$ được định nghĩa như sau:

$$\mathcal{H}(X,Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2 p(x,y)$$

Entropy hợp của hai biến ngẫu nhiên xác định lượng thông tin kết hợp giữa chúng. Đối với hai biến ngẫu nhiên độc lập, entropy hợp là tổng của các entropy thành phần, bởi vì độ bất định của giá trị hai biến sẽ ít hơn so với trường hợp hai biến này độc lập, hay nói cách khác $\mathcal{H}(X,Y) \leq \mathcal{H}(X) + \mathcal{H}(Y)$.

Entropy có điều kiện đo lường sự không dự đoán được của một biến ngẫu nhiên cho trước sao cho giả sử đã biết trước giá trị của một biến ngẫu nhiên khác. Entropy có điều kiện được định nghĩa dựa trên entropy hợp của hai biến ngẫu nhiên và entropy của biến ngẫu nhiên đã biết trước giá trị.

Định nghĩa 1.9 [*Entropy có điều kiện [10]*] Giả sử (X,Y) tuân theo phân bố xác suất hợp $p(x,y)$, xác suất có điều kiện $\mathcal{H}(Y|X) = \sum_x p(x)\mathcal{H}(Y|X=x)$ được xác định như sau:

$$\begin{aligned}\mathcal{H}(Y|X) &= \sum_{x\in X} p(x)\mathcal{H}(Y|X=x)\\ &= -\sum_{x\in X} p(x)\sum_{y\in Y} p(y|x)\log_2 p(y|x) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2 p(y|x)\end{aligned}$$

Để tính toán lượng thông tin của một biến ngẫu nhiên chứa trong giá trị một biến ngẫu nhiên khác, chúng ta sử dụng khái niệm lượng tin tương hỗ. Lượng tin tương hỗ chính là sự giảm độ bất định về một biến ngẫu nhiên khi đã biết về một biến ngẫu nhiên khác.

Định nghĩa 1.10 [*Lượng tin tương hỗ*] Gọi $p(x,y)$ là phân bố xác suất hợp của $x \in X$ và $y \in Y$, khi đó lượng tin tương hỗ giữa X và Y, $\mathcal{I}(X;\ Y)$, được cho bởi công thức sau:

$$\begin{aligned}\mathcal{I}(X;\ Y) &= \sum_{x\in X}\sum_{y\in Y} p(x,y)\log_2\frac{p(x,y)}{p(x)p(y)}\\ &= \mathcal{H}(X) - \mathcal{H}(X|Y) = \mathcal{H}(Y) - \mathcal{H}(Y|X)\end{aligned}$$

1.1.2. Min-entropy

Định nghĩa 1.11 [*Min-entropy [5]*] Gọi $p(x)$ là xác suất của biến X nhận được tại giá trị x, khi đó min-entropy $\mathcal{H}_{Rényi}(X)$ của biến ngẫu nhiên X được định nghĩa như sau:

$$\mathcal{H}_{Rényi}(X) = log_2\frac{1}{\max\limits_{x\in X} p(x)} = -log_2\max_{x\in X} p(x)$$

Như xác định theo công thức trên, chỉ có giá trị phân bố xác suất lớn nhất trong tập phân bố xác suất của biến ngẫu nhiên X được sử dụng để tính toán min-entropy. Điều này có nghĩa là min-entropy chỉ quan tâm đến thành phần có ảnh hưởng lớn nhất đến kết quả.

Định nghĩa 1.12 [*Min-entropy có điều kiện theo Smith [5]*] Min-entropy có điều kiện của một biến ngẫu nhiên X với biến ngẫu nhiên Y cho trước được xác định theo công thức sau:

$$\mathcal{H}_{Smith}(X|Y) = -log_2 \sum_{y \in Y} p(Y = y) \cdot \max_{x \in X} p(X = x|Y = y)$$

Định nghĩa 1.13 [*Min-entropy có điều kiện theo Cachin [7]*] Min-entropy có điều kiện của một biến ngẫu nhiên X với biến ngẫu nhiên Y cho trước được xác định theo công thức sau:

$$\mathcal{H}_{Cachin}(X|Y) = -\sum_{y \in Y} p(Y = y) \cdot log_2 \max_{x \in X} p(X = x|Y = y)$$

CHƯƠNG 2. PHÂN TÍCH ĐỊNH LƯỢNG LUỒNG TIN RÒ RỈ CỦA CHƯƠNG TRÌNH

2.1.1. Phân tích định tính luồng thông tin

Một phương pháp phân tích luồng tin là phương pháp phân tích định tính luồng thông tin. Phương pháp này xác định thông tin bí mật có bị tiết lộ thông qua các thông tin công cộng hay không. Nếu có, hệ thống này sẽ bị xem là không bảo mật.

Luồng thông tin chính là luồng di chuyển của thông tin từ biến này đến biến khác trong chương trình hay hệ thống thông tin. Trong phương pháp phân tích luồng thông tin, mỗi biến sẽ được gán một mức độ bảo mật. Thông thường, một mô hình cơ bản sẽ bao gồm hai mức độ bảo mật: *thấp (low) (L)* và *cao (high) (H)*, hay còn gọi là thông tin công cộng có thể quan sát được (*publicly observable*) và thông tin bí mật (*private*). Phương pháp phân tích định tính luồng thông tin sẽ xác định liệu có bất kỳ luồng thông tin nào từ mức độ bảo mật cao di chuyển đến mức độ bảo mật thấp hay không. Ví dụ với chương trình dưới đây:

$$if\ (S > 0)\ then\ O := 0\ else\ O := 1;$$

trong đó S là biến bí mật, O là biến công cộng, không thoả mãn tính chất định tính bảo mật, vì thông qua giá trị của O, người tấn công có thể biết được thông tin về S.

2.1.2. Phân tích định lượng luồng thông tin

Nếu như mục đích phân tích định tính nhằm xác định một hệ thống có an toàn không bằng cách trả lời câu hỏi rằng hệ thống có bị rò rỉ thông tin hay không, thì phân tích định lượng, ngoài việc xác định luồng di chuyển của thông tin, còn cho phép xác định hay ước lượng lượng tin mà người tấn công có thể thu thập được khi hệ thống bị mất bảo mật. Hay nói cách khác, phân tích định lượng cho phép xác định được có lượng thông tin đã bị tiết lộ ra bên ngoài trong trường hợp hệ thống rò rỉ thông tin. Rõ ràng, phương pháp phân tích định lượng hiệu quả hơn ở khía cạnh bảo mật thông tin. Phương pháp phân tích luồng thông tin có yêu cầu xác định có bao nhiêu lượng thông tin bị tiết lộ được gọi là phương pháp *định lượng luồng thông tin* (*Quantitative Information Flow*).

Như đã trình bày ở trên, tính chất không can nhiễu được sử dụng để chứng minh rằng hệ thống hoạt động an toàn, trong khi tính can nhiễu sẽ chỉ ra một số điểm không hoạt động tốt. Tuy nhiên, điều này chỉ đúng trong trường hợp sự can nhiễu lớn hơn một *mức ngưỡng* (*threshold*) nào đó. Một ví dụ điển hình đó là hệ thống thông tin sử dụng điều khiển truy cập. Để có thể đăng nhập vào hệ thống, người sử dụng cần phải thực hiện bước chứng thực bằng cách sử dụng một khoá mật khẩu. Trong quá trình này, bất kể thông tin chứng thực được nhập vào là đúng hay sai, cũng đã làm rỏ rỉ một phần nhỏ lượng thông tin. Điều này có thể giải thích là dù cho người tấn công sử dụng sai mật khẩu, cũng đã chỉ ra rằng mật khẩu đó không phải là mật khẩu đúng, từ đó làm tiết

lộ một phần nhỏ thông tin bí mật. Trường hợp này, hệ thống đã xuất hiện sự can nhiễu. Nếu sự can nhiễu này là đủ nhỏ, ta xem như hệ thống vẫn đảm bảo được tính bảo mật của nó.

Để quá trình phân tích định lượng luồng tin được rõ ràng và sáng tỏ, chúng ta giả sử rằng: Thứ nhất, chương trình luôn luôn kết thúc, và người tấn công biết về mã nguồn của chương trình. Chúng ta chỉ giới hạn chương trình chỉ có một *đầu vào có độ bảo mật cao S* và một *đầu ra có độ bảo mật thấp O*. Mục tiêu đó là tính toán lượng thông tin về S đã bị thu được bằng cách quan sát O. Chúng ta cũng giả sử rằng tập giá trị của dữ liệu là hữu hạn.

Thứ hai, giả sử rằng có một *phân bố xác suất tiền định, đã biết trước* của các giá trị của biến bí mật.

Cuối cùng, mô hình sử dụng trong phân tích là mô hình *một lần thử* (*one-try guessing*), có nghĩa là sau khi quan sát đầu ra của chương trình, người tấn công chỉ có thể dự đoán giá trị của S một lần duy nhất. Mô hình này phù hợp với nhiều trường hợp an ninh, ví dụ hệ thống sẽ kích hoạt chuông báo động hoặc khoá tài khoản nếu người tấn công bấm sai mật khẩu một lần.

Xét một số ví dụ để hiểu rõ hơn về lượng tin bị tiết lộ khi các thực thi chương trình như bên dưới:

$(P_1)\ O := S;$

$(P_2)\ O := S\ mod\ 2;$

$(P_3)\ if\ (S == 0)\ then\ O := 0\ else\ O := 1;$

trong đó S là biến bí mật, O là biến công cộng.

Chương trình P_1, người tấn công biết hoàn toàn về thông tin bí mật của S. Chương trình P_2 cho biết biến S là chẵn hay lẻ thông qua biến O, hay chương trình đã làm tiết lộ một bit thông tin của S. Tương tự, chương trình P_3 cũng làm một bit thông tin của S di chuyển vào O, và O sẽ biết được biến S có bằng 0 hay không.

Ta nhận thấy rằng quá trình thực thi chương trình đã làm giảm sự không chắc chắn về thông tin bí mật và gây ra sự tiết lộ thông tin. Phương pháp định lượng luồng thông tin sẽ cung cấp công cụ để tính toán lượng thông tin bị tiết lộ, từ đó so sánh với một mức ngưỡng cho trước để xác định chương trình có bảo mật hay không. Các phương pháp tính toán định lượng luồng thông tin chủ yếu dựa trên cơ sở của lý thuyết thông tin Shannon. Để tính toán định lượng thông tin, ta xem chương trình như một kênh thông tin. Khi đó, lý thuyết toán học về thông tin của Shannon sẽ phù hợp để tính toán lượng thông tin qua các kênh truyền này.

2.2. Lượng tin rò rỉ

Phương pháp phân tích định lượng cổ điển sử dụng lý thuyết thông tin để mô hình luồng thông tin và định nghĩa luồng thông tin rò rỉ. Điều cơ bản của các phương pháp này xem chương trình như một kênh thông tin.

Cho một kênh thông tin $\mathcal{M}$ có các thông số như sau $\mathcal{M} = (X, Y, M)$ trong đó X biểu diễn một tập hữu hạn các giá trị bí mật đầu vào, Y biểu diễn một tập hữu hạn các giá trị đầu ra có thể quan sát được, và M là một ma trận kênh $|X| \times |Y|$ chứa các xác suất có điều kiện $p(y|x)$ với mỗi $x \in X$ và $y \in Y$. Mỗi phần tử trong ma trận M là một giá trị thực trong khoảng 0 và 1, và tổng của mỗi hàng sẽ bằng 1.

Về cơ bản, phương pháp phân tích định lượng luồng tin cổ điển mô hình chương trình như một kênh đầu vào-đầu ra tiêu chuẩn với biến bí mật S là đầu vào và O là biến đầu ra công cộng. Phương pháp phân tích sẽ chỉ ra có bao nhiêu thông tin về S mà một người tấn công có thể thu được từ thông tin quan sát được từ O. Hay nói cách khác, đó chính là tính toán lượng tin rò rỉ của chương trình.

Giả sử chương trình P được mô hình như một ma trận kênh với S đầu vào và O đầu ra. Lượng tin rò rỉ của P được định nghĩa bằng hiệu giữa độ bất định mà người tấn công biết về S trước khi thực thi chương trình và độ bất định sau khi quan sát O. Gọi $\mathcal{H}(S)$ là độ bất định ban đầu, và $\mathcal{H}(S|O)$ là độ bất định sau khi chương trình đã được thực thi và các đầu ra đã được quan sát. Lúc này, lượng tin rò rỉ của chương trình được cho bởi,

$$\mathcal{L}(C) = \mathcal{H}(S) - \mathcal{H}(S|O)$$

Trong đó $\mathcal{L}(C)$ là lượng tin rò rỉ của P. $\mathcal{H}$ sẽ được tính hoặc theo định nghĩa của Shannon hoặc định nghĩa của min-entropy. Lượng tin rò rỉ có đơn vị tính bằng bit.

CHƯƠNG 3. THUẬT TOÁN ƯỚC LƯỢNG LUỒNG TIN RÒ RỈ CHO CÁC CHƯƠNG TRÌNH ĐA LUỒNG

3.1. Bảo mật luồng thông tin trong chương trình đa luồng

3.1.1. Chương trình đa luồng

Có rất nhiều các hệ thống yêu cầu tính bảo mật cao hoạt động dựa trên cơ sở của đa luồng tin (multi-threading). Khái niệm đa luồng tin sử dụng để chỉ ra trong các hệ thống này có nhiều tiến trình hoạt động song song đồng thời với nhau. Một số ví dụ về các hệ thống này đó là dịch vụ hướng web (web-based services), cơ sở dữ liệu và hệ điều hành. Cùng với sự phổ biến của các bộ vi xử lý đa nhân, hay các hệ thống song song như bộ xử lý đồ hoạ, đa luồng đang dần trở thành một tiêu chuẩn trong xử lý thông tin. Tuy nhiên, để đảm bảo được vấn đề bảo mật trong chương trình đa luồng là một thách thức thực sự, do dữ liệu trong các chương trình này thường khó dự đoán được trong quá trình thực thi chương trình, và do đó rất khó để dự đoán được người tấn công đã quan sát được gì.

Xét ví dụ về một chương trình đa luồng như bên dưới, với S là biến lưu thông tin bí mật, O là biến lưu thông tin công cộng, trong đó $S \in H$ (tập các biến có mức độ bảo mật cao, hay là thông tin bí mật), và $O \in L$ (tập các biến có mức độ bảo mật thấp, hay là thông tin công cộng có thể quan sát được).

Ví dụ (Chương trình đa luồng)

$O := 0;$

$(\{if\ (O = 1)\ then\ O := S\ else\ skip\} \parallel O := 1;)$

$O := 1;$

Gọi C_1 và C_2 là hai toán hạng ở bên trái và bên phải của toán tử song song $\parallel$. C_1 và C_2 là hai luồng của chương trình đa luồng. Khi thực thi chương trình này, ta thu được vệt chương trình $T_{|O}$ theo biến O, hay chính là các trạng thái tuần tự của O khi thực hiện chương trình, phụ thuộc vào luồng nào được thực hiện trước.

$$T_{|O} = \begin{cases} [0,1,1] & nếu\ C_1\ thực\ hiện\ trước \\ [0,1,S,1] & nếu\ C_2\ thực\ hiện\ trước \end{cases}$$

3.1.2. Tính bảo mật của chương trình đa luồng

Như đã trình bày ở trên, tính chất *không can nhiễu* (*non-interference*) là một tính chất bảo mật cơ bản thường được sử dụng trong các chương trình tuần tự. Tính chất không can nhiễu sẽ chỉ ra rằng chương trình được xem là bảo mật khi một tập các giá trị cuối cùng có thể có của các biến

công cộng là độc lập với tập các giá trị ban đầu của các biến bí mật. Tuy nhiên, tính chất này lại không phù hợp với chương trình đa luồng. Có hai lý do để giải thích điều này.

Thứ nhất, do sự trao đổi các kết quả trung gian trong quá trình thực thi chương trình đa luồng, việc phân tích luông thông tin cần phải xem xét them sự rò rỉ thông tin ở các trạng thái trung gian này.

Xét lại *Ví dụ trên (chương trình đa luồng),* chương trình được xem là bảo mật, vì giá trị cuối cùng của O là độc lập với giá trị ban đầu của S. Tuy nhiên, chương trình này rò rỉ toàn bộ thông tin bí mật, vì người tấn công có thể truy cập S thông qua trạng thái trung gian của vệt dữ liệu công cộng khi luồng C_2 được thực hiện trước.

Do đó, định nghĩa về tính chất không can nhiễu chỉ xét đến rò rỉ thông tin ở trạng thái cuối là chưa đủ để đảm bảo tính bảo mật của chương trình đa luồng. Trong trường hợp này, cần phải đảm bảo rằng các dữ liệu bí mật cũng không bị tiết lộ trong toàn bộ tiến trình thực thi chương trình đa luồng, hay chính là các trình tự của các trạng thái trong quá trình thực thi chương trình.

Thứ hai, một chương trình đa luồng thực thi các luồng từ một tập các luồng không kết thúc. Trong quá trình thực thi này, bộ lập lịch sẽ lựa chọn lập đi lặp lại luồng nào sẽ được thực thi tiếp theo với một xác suất xác định. Do đó, vệt dữ liệu của chương trình đa luồng sẽ phụ thuộc vào bộ lập lịch được áp dụng trong chương trình. Trong trường hợp này, chúng ta sử dụng khái niệm *chương trình có tính xác suất* (*probabilistic programs*), trong đó giả sử chúng ta biết trước về xác suất của bộ lập lịch khi lựa chọn các luồng.

3.1.3. Ảnh hưởng của bộ lập lịch trong chương trình đa luồng

Vì đầu ra của chương trình đa luồng phụ thuộc vào bộ lập lịch, do đó để có thể xây dựng một mô hình phù hợp cho chương trình đa luồng, chúng ta cần phải biết được người tấn công đã thu được những thông tin gì dựa trên bộ lập lịch được lựa chọn. Ta xem xét ví dụ bên dưới.

Ví dụ:

$$O := S/2 \parallel O := S \bmod 2$$

Giả sử người tấn công thực thi chương trình trên với bộ lập lịch có xác suất phân bố đều, nghĩa là bộ lập lịch sẽ chọn một trong hai luồng để thực hiện với xác suất là bằng nhau. Nếu S có kiểu dữ liệu 2-bit có phân bố đều, sẽ có 4 vệt chương trình có thể có với cùng một xác suất xuất hiện, đó là $\{00,01,10,11\}$.

S	0		1		2		3	
$T_{\|O}$	0 0	0 0	0 1	1 0	1 0	0 1	1 1	1 1

Nếu vệt chương trình thu được là 00 hoặc 11, người tấn công biết chính xác được S sẽ tương ứng bằng 0, hoặc 3. Tuy nhiên, nếu vệt chương trình bằng 01 hoặc 10, người này cũng chỉ có thể dự đoán được S hoặc bằng 1 hoặc bằng 2, với cùng một xác suất chính xác. Do đó, với bộ lập lịch này, thông tin bí mật không bị tiết lộ hoàn toàn.

Tuy nhiên, nếu người tấn công sử dụng bộ lập lịch luôn luôn chọn luồng $O := S/2$ để thực hiện trước, sẽ có 4 vệt chương trình có thể có là $\{00,01,10,11\}$. Tuy nhiên trong trường hợp này, người tấn công luôn biết chính xác được giá trị của S.

Vì vậy, để xây dựng một mô hình phân tích định lượng luồng tin cho chương trình đa luồng, ta cần phải xem xét một số yếu tố sau: (1) giá trị của biến công cộng ở các trạng thái trung gian có ảnh hưởng như thế nào đến dữ liệu bí mật mà người tấn công có thể thu được, (2) ảnh hưởng của bộ lập lịch đến lượng thông tin rò rỉ của chương trình.

3.1.4. Mô hình chương trình đa luồng

Mô hình chương trình đa luồng được xây dựng dựa trên bộ lập lịch có tính xác suất.

Định nghĩa 3.1 (Mô hình chương trình đa luồng) Mô hình chương trình được biểu diễn dưới dạng $\mathcal{A} = \langle \mathcal{S}, \mathcal{I}, Var, \mathcal{V}, \rightarrow \rangle$**,** trong đó:

1. $\mathcal{S}$ là tập trạng thái có thể có trong chương trình,
2. $\mathcal{I} \in \mathcal{S}$ là trạng thái đầu tiên.
3. $Var = H \cup L$ là tập hữu hạn biến của chương trình đa luồng.
4. Hàm nhãn $\mathcal{V} : \mathcal{S} \rightarrow \mathcal{D}(S)$, với $S \in H$, đánh nhãn các trạng thái với phân bố xác suất của biến bí mật S, nghĩa là ánh xạ phân bố xác suất $\mu \in \mathcal{D}(S)$ đến từng trạng thái $c \in \mathcal{S}$, mô tả thông tin mà người tấn công biết được về S tại mỗi trạng thái.
5. $\rightarrow$ chỉ mối quan hệ chuyển trạng thái, $\rightarrow \subseteq \mathcal{S} \rightarrow \mathcal{D}(S)$.

Phân bố xác suất của S thay đổi từ trạng thái này sang trạng thái tiếp theo dọc theo đường thực thi chương trình, phụ thuộc vào giá trị của biến công cộng ở các trạng thái và lệnh của chương trình.

Xét chương trình bên dưới:

$$O := S/2 \parallel O := S \bmod 2$$

Giả sử người tấn công biết trước được các giá trị của S trong khoảng $\{0, 1, 2, 3\}$, và S có phân bố xác suất đều: $\pi = \left\{0 \mapsto \frac{1}{4}, 1 \mapsto \frac{1}{4}, 2 \mapsto \frac{1}{4}, 3 \mapsto \frac{1}{4}\right\}$. Với bộ lập lịch lựa chọn giữa hai luồng để thực thi trước với xác suất là như nhau, và cùng đều có chung một kết quả của O, với $O = 1$.

Nếu luồng $O \coloneqq S/2$ được thực thi trước, ta có phân bố xác suất được cập nhật lại sẽ là $\left\{0 \mapsto 0, 1 \mapsto 0, 2 \mapsto \frac{1}{2}, 3 \mapsto \frac{1}{2}\right\}$. Ngược lại, nếu luồng $O \coloneqq S \; mod \; 2$ được thực thi trước, phân bố xác suất lúc này sẽ là $\left\{0 \mapsto 0, 1 \mapsto \frac{1}{2}, 2 \mapsto 0{,}3 \mapsto \frac{1}{2}\right\}$.

Ta nhận thấy rằng chương trình đã có sự biến đổi về phân bố xác suất, từ phân bố xác suất ban đầu của S ở trạng thái đầu tiên đến phân bố xác suất cuối ở trạng thái kết thúc. Bằng cách quan sát sự thay đổi của phân bố xác suất này, người tấn công có thể biết được thông tin về dữ liệu bí mật ban đầu. Mục tiêu trong phân tích định lượng luồng thông tin chính là tính toán, đo đạc được lượng thông tin bí mật mà người tấn công đã thu được dựa trên những quan sát đó.

3.2. Lượng tin rò rỉ của chương trình đa luồng

3.2.1. Lượng tin rò rỉ theo vệt chương trình

Ví dụ dưới đây chỉ ra cách tính lượng tin rò rỉ theo vệt chương trình.

Ví dụ

$O \coloneqq 0;$

$O \coloneqq S \;\&\; (001)_b$

$O \coloneqq S \;\&\; (011)_b$

Gọi $(s_1 s_2 s_3)_b$ biểu diễn giá trị nhị phân của S. Giả sử phân bố ban đầu của S là phân bố đều. Quá trình thực thi chương trình sẽ được một vệt chương trình như sau, $\langle (000)_b \rangle \longrightarrow \langle (00 s_1)_b \rangle \longrightarrow \langle (0 s_2 s_1)_b \rangle$, trong đó $\langle \; \rangle$ biểu diễn giá trị của biến công cộng O.

Giả sử rằng $s_2 = 1$ và $s_1 = 1$, vệt chương trình thu được sẽ trở thành

$$\langle (000)_b \rangle \longrightarrow \langle (001)_b \rangle \longrightarrow \langle (011)_b \rangle$$

Tại trạng thái ban đầu $\langle (000)_b \rangle$, độ bất định của người tấn công được biểu diễn bởi phân bố xác suất của S, $\left\{0 \mapsto \frac{1}{8}, 1 \mapsto \frac{1}{8}, 2 \mapsto \frac{1}{8}, 3 \mapsto \frac{1}{8}, 4 \mapsto \frac{1}{8}, 5 \mapsto \frac{1}{8}, 6 \mapsto \frac{1}{8}, 7 \mapsto \frac{1}{8}\right\}$.

Tại trạng thái $\langle (001)_b \rangle$, người tấn công sẽ biết được bit cuối cùng của S là 1, hay S là không chẵn. Lúc này, phân bố xác suất được cập nhật sẽ là $\left\{0 \mapsto 0, 1 \mapsto \frac{1}{4}, 2 \mapsto 0, 3 \mapsto \frac{1}{4}, 4 \mapsto 0, 5 \mapsto \frac{1}{4}, 6 \mapsto 0, 7 \mapsto \frac{1}{4}\right\}$.

Tương tự, đối với trạng thái $\langle(011)_b\rangle$, phân bố xác suất được cập nhật lại sẽ là $\left\{3 \mapsto \frac{1}{2}, 7 \mapsto \frac{1}{2}\right\}$.

Dựa vào phân bố xác suất cuối cùng của S, người tấn công sẽ biết được giá trị của S là 3 hoặc 7. Độ bất định lúc này đã giảm xuống dựa vào những giá trị biến công cộng quan sát được. Vì chương trình có sự biến đổi phân bố xác suất, trong đó phân bố xác suất của biến bí mật tại trạng thái ban đầu biểu diễn cho độ bất định ban đầu của người tấn công về thông tin bí mật, và phân bố xác suất của biến bí mật ở trạng thái cuối cùng biểu diễn độ bất định còn lại, sau khi đã quan sát chương trình được thực thi. Từ đó, lượng tin rò rỉ của chương trình có thể tính bằng:

Lượng tin rò rỉ = Độ bất định ban đầu – Độ bất định còn lại.

Tính độ bất định. Với một phân bố xác suất của bí mật cho trước, trong mô hình dự đoán cho một lần thử, phương án tốt nhất chính là chọn giá trị với xác suất lớn nhất, có nghĩa là xác suất cho việc dự đoán sai sẽ nhỏ nhất. Gọi **S** là tập giá trị có thể của S, giá trị ảnh hưởng đến độ bất định là $max_{s\in \mathbf{S}} p(s)$. Nếu $max_{s\in \mathbf{S}} p(s) = 1$, mức độ bất định phải bằng 0, lúc này người tấn công biết hoàn toàn về giá trị của S. Do đó, biểu thức tính độ bất định sẽ bằng trừ logarithm của $max_{s\in \mathbf{S}} p(s)$, hay độ bất định $= -log_2 max_{s\in \mathbf{S}} p(s)$, trong đó dấu "-" biểu thị cho sự không âm của độ bất định. Biểu thức này hoàn toàn phù hợp với công thức tính min-entropy của Rényi.

Định nghĩa 3.2. Gọi **X** là tập hữu hạn các giá trị của biến ngẫu nhiên X. Biểu thức tính min-entropy Rényi của X được định nghĩa bởi:

$$\mathcal{H}_{Rényi}(X) = log_2 \frac{1}{\max\limits_{x\in X} p(x)} = -log_2 \max\limits_{x\in X} p(x)$$

Với phân bố xác suất S cho trước, độ bất định của người tấn công về thông tin bí mật sẽ là: Độ bất định $= \mathcal{H}_{Rényi}(S)$.

Do vậy, lượng thông tin rò rỉ theo vệt chương trình T, là $\mathcal{L}(T)$, được tính theo công thức,

$$\mathcal{L}(T) = \mathcal{H}_{Rényi}\left(S_T^i\right) - \mathcal{H}_{Rényi}\left(S_T^f\right)$$

trong đó $\mathcal{H}_{Rényi}\left(S_T^i\right)$ là min-entropy của S ứng với phân bố xác suất ban đầu, và $\mathcal{H}_{Rényi}\left(S_T^f\right)$ là min-entropy tương ứng với phân bố xác suất cuối cùng, tại thời điểm trạng thái cuối cùng khi kết thúc chương trình.

3.2.2. Lượng tin rò rỉ của chương trình đa luồng

Quá trình thực thi chương trình đa luồng C dưới sự điều khiển của bộ lập lịch δ sẽ có kết quả là một tập các vệt chương trình $Trace(\mathcal{A}_\delta)$. Do đó, lượng tin rò rỉ của C sẽ là giá trị rò rỉ tính trên tất cả các vệt chương trình.

$$\mathcal{L}(C,\pi) = \sum_{T \in Trace(\mathcal{A}_\delta)} p(T) \cdot \mathcal{L}(T)$$
$$= \sum_{T \in Trace(\mathcal{A}_\delta)} p(T) \cdot \left(\mathcal{H}_{Rényi}(S_T^i) - \mathcal{H}_{Rényi}(S_T^f)\right)$$

Vì $\mathcal{H}_{Rényi}(S_T^i)$ là như nhau đối với mọi $T \in Trace(\mathcal{A}_\delta)$, ta viết lại là $\mathcal{H}_{Rényi}(S^i)$. Biểu thức trên trở thành,

$$\mathcal{L}(C,\pi) = \mathcal{H}_{Rényi}(S^i) - \sum_{T \in Trace(\mathcal{A}_\delta)} p(T) \cdot \mathcal{H}_{Rényi}(S_T^f)$$

với $p(T)$ là xác suất tích luỹ dọc theo vệt chương trình T.

3.2.3. Ví dụ minh hoạ

Ví dụ dưới đây minh hoạ phương pháp tính lượng tin rò rỉ của chương trình đa luồng. Xét chương trình như bên dưới, trong đó S là biến kiểu nguyên không dấu 3-bit,

$O \coloneqq 0;$

$(if\ (O = 1)\ then\ O \coloneqq S/4\ \ else\ O \coloneqq S\ mod\ 2)\ \|\ O \coloneqq 1;$

$O \coloneqq S\ mod\ 4;$

Quá trình thực thi chương trình này, với bộ lập lịch phân bố đều, ta thu được 20 trạng thái được đánh số từ **0** (trạng thái ban đầu) đến **19**. Nội dung trong mỗi trạng thái là giá trị của biến O ở trạng thái đó. Ví dụ, ở trạng thái ban đầu, giá trị của O là 0, tương ứng với dòng lệnh đầu tiên của chương trình $O \coloneqq 0$.

Gọi C_1 và C_2 lần lượt là luồng chương trình bên trái và bên phải. Vì áp dụng bộ lập lịch phân bố đều, do đó việc lựa chọn C_1 hay C_2 để thực thi trước sẽ có cùng xác suất là $\frac{1}{2}$. Nếu bộ lập lịch lựa chọn C_2 thực hiện trước C_1, trạng thái **0** sẽ chuyển sang trạng thái **1**, với $O = 1$. Ngược lại, nếu C_1 thực hiện trước, trạng thái **0** sẽ chuyển sang trạng thái **2** hoặc **3** với cùng xác suất là $\frac{1}{4}$. Vì giá trị của O ở trạng thái **0** là 0, câu lệnh $O \coloneqq S\ mod\ 2$ sẽ được thực hiện. Vì tập giá trị của S là $\{0, 1, 2, 3, 4, 5, 6, 7\}$, giá trị của O có thể là 0 (trạng thái **2**) nếu $S \in \{0, 2, 4, 6\}$, hoặc 1 (trạng thái **3**) nếu $S \in \{1, 3, 5, 7\}$.

Tại trạng thái **1**, chương trình có thể chuyển sang trạng thái **4** hoặc trạng thái **5** với cùng một xác suất. Lúc này, vì $O = 1$, nên câu lệnh $O \coloneqq S/4$ sẽ được thực hiện. Do đó, giá trị của O có thể là 0 nếu $S \in \{0, 1, 2, 3\}$, hoặc 1 nếu $S \in \{4, 5, 6, 7\}$.

Chương trình kết thúc khi câu lệnh cuối cùng được thực hiện, $O \coloneqq S\ mod\ 4$.

Tại trạng thái ban đầu, độ bất định của người tấn công về biến S tuân theo phân bố xác suất như sau,

$$\pi = \left\{0 \mapsto \frac{1}{8}, 1 \mapsto \frac{1}{8}, 2 \mapsto \frac{1}{8}, 3 \mapsto \frac{1}{8}, 4 \mapsto \frac{1}{8}, 5 \mapsto \frac{1}{8}, 6 \mapsto \frac{1}{8}, 7 \mapsto \frac{1}{8}\right\}$$

Tại trạng thái **1**, phân bố xác suất của S vẫn không thay đổi, do người tấn công vẫn chưa thu được thông tin thông qua câu lệnh $O \coloneqq 1$. Tại trạng thái **4**, kết quả của câu lệnh $O \coloneqq S/4$ là 0, người tấn công biết được giá trị thực của S sẽ nằm trong tập $\{0, 1, 2, 3\}$, với xác suất giống nhau. Do đó, phân bố xác suất cập nhật của S ở trạng thái này sẽ là: $\left\{0 \mapsto \frac{1}{4}, 1 \mapsto \frac{1}{4}, 2 \mapsto \frac{1}{4}, 3 \mapsto \frac{1}{4}\right\}$.

Tiếp theo, khi câu lệnh $O \coloneqq S \; mod \; 4$ được thực hiện, người tấn công biết giá trị S chính xác hơn. Ví dụ, tại trạng thái **8**, vì $O = 0$, phân bố xác suất của S là: $\{0 \mapsto 1\}$. Tương tự đối với các trạng thái **9**, …, **15**, người tấn công cũng sẽ biết được giá trị chính xác của S dựa trên phân bố xác suất cuối cùng.

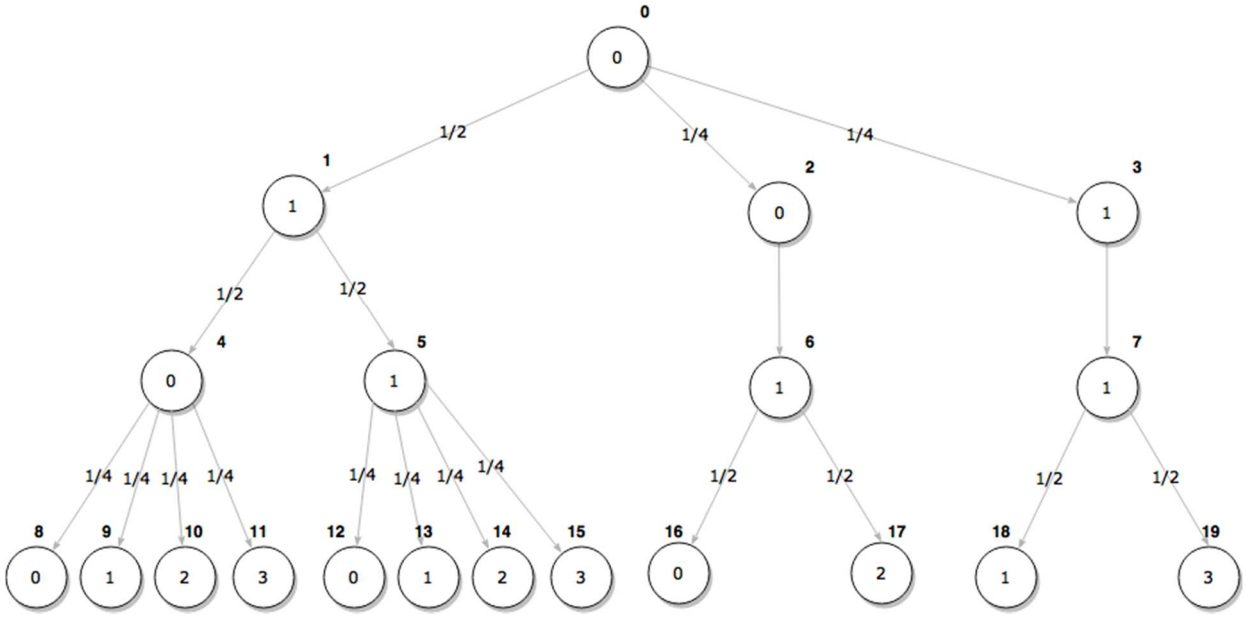
Ở trạng thái **2**, kết quả của câu lệnh $O \coloneqq S \; mod \; 2$ sẽ cho kết quả $O = 0$, phân bố xác suất ở trạng thái này sẽ là: $\left\{0 \mapsto \frac{1}{4}, 2 \mapsto \frac{1}{4}, 4 \mapsto \frac{1}{4}, 6 \mapsto \frac{1}{4}\right\}$. Phân bố xác suất không thay đổi ở trạng thái **6**, do thực thi câu lệnh $O \coloneqq 1$ không thu được thêm thông tin nào. Tại trạng thái **16**, cập nhật phân bố của S sẽ là: $\left\{0 \mapsto \frac{1}{4}, 4 \mapsto \frac{1}{4}\right\}$, do kết quả của câu lệnh $O \coloneqq S \; mod \; 4$ bằng 0. Tương tự với trạng thái **17**, **18**, **19**, ta cũng có được phân bố xác suất có dạng tương tự.

Chương trình kết thúc với 12 vệt chương trình. Trong tổng số 12 vệt chương trình này, có 8 vệt chương trình có độ bất định cuối cùng bằng 0, hay $\mathcal{H}_{Rényi}\left(S_T^f\right) = -\log_2 1 = 0$, và 4 vệt chương trình còn lại có độ bất định bằng 1, hay $\mathcal{H}_{Rényi}\left(S_T^f\right) = -\log_2 \frac{1}{2} = 1$.

Xác suất xảy ra vệt chương trình với độ bất định bằng 0 bằng với xác suất xảy ra vệt chương trình với độ bất định bằng 1. Do đó, theo phân tích như trên, lượng tin rò rỉ sẽ là,

$$\mathcal{L}(C, \pi) = 3 - \left(\frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 1\right) = 2.5 \; (bits).$$

Giá trị tính toán này phù hợp với lượng tin rò rỉ thực của chương trình. Rõ ràng đối với câu lệnh $O \coloneqq S \; mod \; 4$ sẽ luôn làm tiết lộ 2 bits thông tin của S. Bit đầu tiên có thể bị tiết lộ với xác suất $\frac{1}{2}$, tuỳ thuộc vào bộ lập lịch lựa chọn thực hiện luồng C_2 trước hay không. Chính vì vậy, đối với bộ lập lịch có phân bố đều, lượng tin rò rỉ thực sự sẽ là 2.5 bits.



Hình 1. Cây trạng thái chương trình của ví dụ

CHƯƠNG 4. CHƯƠNG TRÌNH MÔ PHỎNG PHÂN TÍCH ĐỊNH LƯỢNG LUỒNG TIN

4.1. Tổng quan chương trình mô phỏng

Đề tài giới thiệu một phương pháp xây dựng chương trình mô phỏng tính toán lượng thông tin rò rỉ của chương trình đa luồng. Chương trình mô phỏng được xây dựng dựa sự biến đổi phân bố xác suất của biến thông tin bí mật và tính toán lượng thông tin rò rỉ theo phân bố xác suất được cập nhật.

Dựa trên phân bố xác suất ban đầu, tập giá trị của biến bí mật, và mã nguồn của chương trình đa luồng, chương trình mô phỏng sẽ thực hiện từng câu lệnh, tính toán cập nhật lại phân bố xác suất, hình thành không gian trạng thái của chương trình. Từ đó, công cụ sẽ dựa trên không gian trạng thái và phân bố xác suất cuối cùng để tính toán lượng thông tin rò rỉ. Đầu ra của chương trình mô phỏng này lượng thông tin rò tỉ tại thời điểm kết thúc của chương trình.

Chương trình mô phỏng tính toán lượng thông tin rò rỉ được lập trình trên phần mềm MATLAB®. Tất cả cú pháp câu lệnh, kiểu dữ liệu, các biểu thức, hàm tuân theo ngôn ngữ lập trình MATLAB.

4.2. Cấu trúc chung của chương trình mô phỏng

Cấu trúc công cụ gồm ba phần chính: Đầu vào (Input), Tính toán mô phỏng (Analyser), Đầu ra (Output).

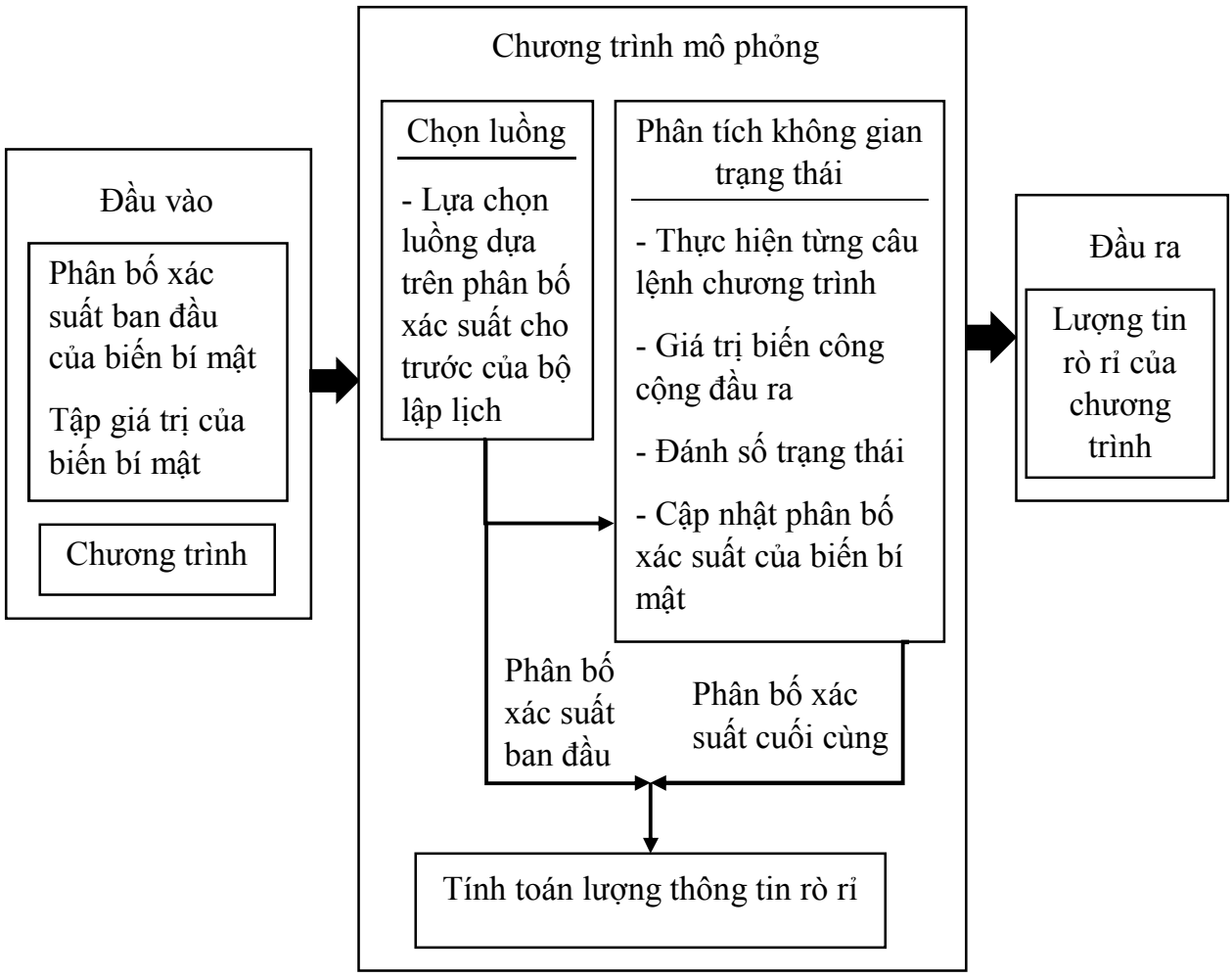
Đầu vào (Input) chứa các khai báo đầu vào của chương trình bao gồm:

- Khai báo biến bí mật, biến công cộng
- Khai báo tập giá trị của biến bí mật
- Khởi tạo giá trị cho biến công cộng
- Khởi tạo cấu trúc không gian trạng thái
- Chương trình đa luồng để thực thi

Các giá trị cho biến bí mật, biến công cộng, và phân bố xác suất ban đầu được khởi tạo dựa vào yêu cầu đầu vào của chương trình đã cho. Không gian trạng thái lưu trữ các trạng thái tương ứng trong quá trình thực thi chương trình. Không gian trạng thái là một mảng có n phần tử có dữ liệu kiểu *struct*, số phần tử này tương ứng với số trạng thái có được khi thực thi chương trình, được mô tả như sau:

$$\langle state\ i \rangle = \begin{cases} state \\ observe \\ secret \\ probDist \\ probTrace \end{cases}$$

trong đó, trạng thái thứ i $\langle state\ i \rangle$ chứa các nội dung như sau: (1) $state$ là số thứ tự của trạng thái, (2) $observe$ chứa giá trị của biến quan sát (biến công cộng) tại trạng thái i, (3) $secret$ chứa tập giá trị của biến bí mật tương ứng, (4) $probDist$ chứa phân bố xác suất đã được cập nhật lại dựa trên tập giá trị của biến $secret$, (5) $probTrace$ lưu xác suất chuyển trạng thái tích luỹ.



Hình 2. Cấu trúc chương trình mô phỏng

Quá trình thực thi chương trình để khám phá không gian trạng thái được bắt đầu bằng quá trình lựa chọn luồng để thực thi. Chương trình mô phỏng sẽ thực hiện vòng lặp với số lần lặp nhất định để đảm bảo các trường hợp thực thi luồng tin sẽ được khai phá đầy đủ.

Tại quá trình khám phá không gian trạng thái, từng câu lệnh sẽ được thực hiện. Dựa trên kết quả của câu lệnh này, cụ thể là các giá trị biến công cộng, chương trình sẽ tính toán và cập nhật giá trị biến bí mật và phân bố xác suất tương ứng.

Sau khi đã thu được toàn bộ không gian trạng thái tại thời điểm kết thúc chương trình đa luồng cần phân tích, chương trình mô phỏng sẽ thực hiện tính toán lượng tin rò rỉ dựa trên độ bất định ban đầu và độ bất định cuối cùng, theo công thức như đã trình bày ở chương 3. Từ đó, chương trình mô phỏng sẽ hiển thị lượng tin rò rỉ của chương trình đa luồng.