Proposta de Projeto Orientado em Computação I

Artur Duarte Penna Vaz
Orientador: Mário S Alvim

Universidade Federal de Minas Gerais, Brazil

1 Introdução

Proteger informação sensível do público é um objetivo importante de segurança. O campo do Fluxo Quantitativo de Informação(QIF) se preocupa em quantificar quanto de informação sensível um sistema vaza, e tem sido muito ativo na última década.

A representação do sistema é chamada de *Canal* e é a distribuição de probabilidade das saídas de cada entrada, essa definição modela o comportamento do sistema.

Em sistemas complexos derivar o canal diretamente não é trivial, ao contrario de sistemas pequenos/simples. A partir disso, foi proposto uma abordagem aproximativa que modela o canal da composição de partes do sistema e é proposto operadores que capturam as interações que ocorrem entre os componentes. Essa abordagem simplifica o processo de modelagem.

Com a orientação do professor Mário S. Alvim, o objetivo desse projeto é comparar a análise de fluxo de informação pela aproximação usando os operadores e o calculo exato no canal, e modelar novos protocolos como composição de canais e analisar sua vulnerabilidade. Na primeira parte do Projeto Orientado em Computação será feita uma comparação do vazamento de informação em protocolos já estudados, usando as diferentes abordagens, e será implementada uma biblioteca em C++ para auxiliar esse projeto, e futuros. Na segunda parte novos protocolos serão analisados usando o método aproximativo.

2 Referencial Teórico

No campo de QIF, sistemas são modelados como canais de informação. Um canal é definido como uma função $C : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ onde $\mathcal{X}$ é o grupo das entradas, ou valores secretos, e $\mathcal{Y}$ é grupo das saídas, ou observáveis:

$$C(x, y) = p(y|x)$$

$C(x, y)$ é a probalidade do sistema produzir o observável y dado o segredo x, $\forall x \in \mathcal{X}$ e $\forall y \in \mathcal{Y}$.

Por exemplo, podemos modelar um sistema de *login* onde $\mathcal{X}$ é o grupo de todas as possíveis senhas e $\mathcal{Y}$ um grupo de 2 elementos, se a senha é correta ou incorreta. Nesse sistema apenas um elemento é correto e o restante incorreto. O

C	Correto	Incorreto
”123456”	1	0
Restante	0	1

Table 1: Representação matricial de um canal

canal do login(Table 1) é simples e interessante para perceber a interpretação do

canal. Em sistemas de segurança é comum ter varias observáveis possíveis para cada segredo.

Para entender como a informação vaza é preciso modelar também o *atacante*, que vê o observável e conhece o funcionamento do sistema. Alem disso, o adversário pode saber alguma coisa sobre o segredo antes do sistema executar, definido como distribuição a priori π em $\mathcal{X}$.

Para quantificar a vulnerabilidade do sistema muitas funções foram usadas na literatura, como a *Shannon Entropy*[8], *Guessing Entropy*[6], *Bayes Vunerability*[4] e *Rényi min-entropy*[9]. Recentemente, *g-leakage*[2] foi proposto e teve muito sucesso em generalizar as funções anteriores e capturar novos cenários.

Nessa estrutura, o grupo de ações possíveis $\mathcal{W}$ e uma função de ganho $g : \mathcal{W} \times \mathcal{X} \to [0, 1]$ definem o ganho do adversário ao escolher uma ação $w \in \mathcal{W}$ dado o segredo $x \in \mathcal{X}$. Dado uma função de ganho g, a *g-vunerability* a priori é:

$$V_g[\pi] = \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi(x) g(w, x)$$

A *g-vunerability* a priori mede o ganho esperado do adversário baseado no seu conhecimento prévio sobre o segredo se tomar a melhor decisão. Precisamos definir também a vulnerabilidade posterior:

$$\begin{aligned} V_g[\pi\rangle C] &= \sum_{y \in \mathcal{Y}} \max_{w \in \mathcal{W}} \sum_{x \in \mathcal{X}} \pi(x) g(w, x) C(x, y) \\ &= \sum_{y \in \mathcal{Y}} p(y) V_g(p(X|_y) \end{aligned}$$

O vazamento de informação é o aumento da vulnerabilidade do segredo causado pela observação do output do sistema. Vazamento é definido pela comparação da vulnerabilidade posterior e a priori do segredo, existe a noção multiplicativa e aditiva[1]:

$$\mathcal{L}_g^{\times}[\pi\rangle C] = \frac{V_g[\pi\rangle C]}{V_g[\pi]} \qquad\qquad \mathcal{L}_g^{+}[\pi\rangle C] = V_g[\pi\rangle C] - V_g[\pi]$$

Em[3] é definido operadores que capturam a interação entre os componentes e uma modelagem do protocolo *Dining Cryptographers* é apresentada.

Operador paralelo $\parallel$

O operador paralelo modela a composição de dois canais que recebem a mesma entrada e as respectivas saídas observadas.

Definição: Dado *canais compatíveis* $C_1 : \mathcal{X} \times \mathcal{Y}_1 \to \mathbb{R}$ e $C_2 : \mathcal{X} \times \mathcal{Y}_2 \to \mathbb{R}$, sua composição paralela $C_1 \parallel C_2 : \mathcal{X} \times (\mathcal{Y}_1 \times \mathcal{Y}_2) \to \mathbb{R}$, para todo $x \in \mathcal{X}, y_1 \in \mathcal{Y}_1, y_2 \in \mathcal{Y}_2$, temos:

$$(C_1 \parallel C_2)(x, (y_1, y_2)) = C_1(x, y_1) \cdot C_2(x, y_2).$$

Operador visible choice $_p\llcorner\lrcorner$

O operador *visible choice* modela a escolha de qual canal executar a entrada, cada canal recebe uma probabilidade de executar. É retornado a saída e um identificador do canal executado.

Definição: Dado *canais compativeis* $C_1 : \mathcal{X} \times \mathcal{Y}_1 \to \mathbb{R}$ e $C_2 : \mathcal{X} \times \mathcal{Y}_2 \to \mathbb{R}$, a *hidden choice* é o *canal* $C_1{}_p\llcorner\lrcorner C_2 : \mathcal{X} \times (\mathcal{Y}_1 \sqcup \mathcal{Y}_2) \to \mathbb{R}$, para todo $x \in \mathcal{X}$ e $(y, i) \in \mathcal{Y}_1 \sqcup \mathcal{Y}_2$,

$$(C_1{}_p\llcorner\lrcorner C_2)(x,(y,i)) = \begin{cases} p \cdot C_1(x,y), & \textit{se } i = 1 \\ (1-p) \cdot C_2(x,y) & \textit{se } i = 2 \end{cases}$$

Operador hidden choice $_p\oplus$

O operador *hidden choice* é semelhante ao visível porem é retornado a saída sem um identificador.

Definição: Dado *canais compatíveis* $C_1 : \mathcal{X} \times \mathcal{Y}_1 \to \mathbb{R}$ e $C_2 : \mathcal{X} \times \mathcal{Y}_2 \to \mathbb{R}$, a *hidden choice* é o *canal* $C_1{}_p\oplus C_2 : \mathcal{X} \times (\mathcal{Y}_1 \cup \mathcal{Y}_2) \to \mathbb{R}$, para todo $x \in \mathcal{X}$ e $(y, i) \in \mathcal{Y}_1 \cup \mathcal{Y}_2$,

$$(C_1{}_p\oplus C_2)(x,y) = \begin{cases} p \cdot C_1(x,y) + (1-p) \cdot C_2(x,y), & \textit{se } y \in \mathcal{Y}_1 \cap \mathcal{Y}_2 \\ p \cdot C_1(x,y) & \textit{se } y \in \mathcal{Y}_1 \setminus \mathcal{Y}_2, \\ (1-p) \cdot C_2(x,y), & \textit{se } y \in \mathcal{Y}_2 \setminus \mathcal{Y}_1. \end{cases}$$

3 Metodologia

O projeto consiste em implementar as ferramentas e analisar a diferença entre a aproximação do vazamento, usando os operadores, e o cálculo exato feito diretamente no canal.

A primeira parte do projeto é implementar uma biblioteca em C++ que contenha o ferramental para análise. Essa biblioteca vai conter:

- Estrutura do canal
- Utilidades para a classe do canal
- Métricas antigas
- Métricas novas
- Operadores de composição

O canal vai ser definido como uma classe onde as métricas e os operadores são métodos. Será implementado algumas utilidades, por exemplo, gerar um canal aleatório, verificar se canais são compatíveis, etc.

A segunda parte é analisar a efetividade dos operadores de composição. Existem dois protocolos de segurança, *Dining Cryptographers*[5] e *Crowds*[7], comuns na literatura de QIF que podem ser usados para comparar o vazamento de informação usando os métodos diferentes.

4 Resultados Esperados

Ao final do POC I teremos uma implementação para o framework de QIF, com métricas e operações, que por si só já em um boa contribuição para futuras pesquisas. Além disso, mostrar que métodos aproximativos são uma alternativa viável para modelar sistemas de segurança contribui para a modelagem de sistemas cada vez maiores e complexos.

No POC II é esperado expandir a quantidade de sistemas de segurança modelados e analisados na literatura.

5 Etapas e Cronograma

O cronograma(Fig. 1) foi definido dividindo as etapas em blocos de semana, e pelo planejado a implementação e a análise deve estar pronta na época da apresentação parcial.

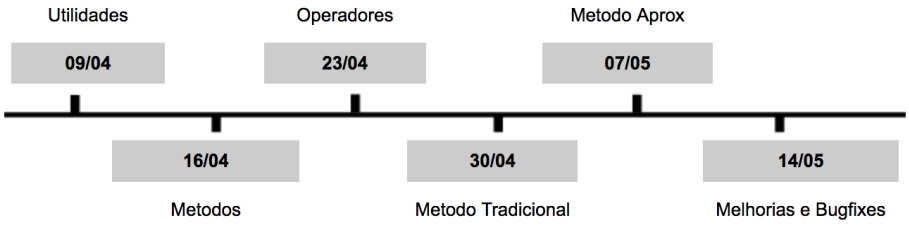

Fig. 1: Cronograma

Do dia 14/05 para frente será reservado para melhorar a usabilidade da biblioteca e resolver eventuais problemas.

Referências

1. M. S. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, C. Palamidessi, and G. Smith. Additive and multiplicative notions of leakage, and their capacities. In *Proc. of CSF*, pages 308–322. IEEE, 2014.
2. M. S. Alvim, K. Chatzikokolakis, C. Palamidessi, and G. Smith. Measuring information leakage using generalized gain functions. In *Proc. of CSF*, pages 265–279, 2012.
3. A. M. Arthur Américo, Mário S. Alvim. An algebraic approach for reasoning about information flow. *arXiv preprint arXiv:1801.08090*, (3), 2018.
4. C. Braun, K. Chatzikokolakis, and C. Palamidessi. Quantitative notions of leakage for one-try attacks. In *Proc. of MFPS*, volume 249 of *ENTCS*, pages 75–91. Elsevier, 2009.
5. D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

6. Massey. Guessing and entropy. In *Proceedings of the IEEE Int. Symposium on Information Theory*, page 204. IEEE, 1994.
7. M. K. Reiter and A. D. Rubin. Crowds: anonymity for Web transactions. *ACM Trans. on Information and System Security*, 1(1):66–92, 1998.
8. C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 625–56, 1948.
9. G. Smith. On the foundations of quantitative information flow. In *Proc. of FOSSACS*, volume 5504 of *LNCS*, pages 288–302. Springer, 2009.