Quantitative Analysis of Leakage of Confidential Information

David Clark (Kings College)

Sebastian Hunt (City University)

Pasquale Malacaria (Queen Mary)

Dagstuhl October 5–10 2003

Motivation

- Non-interference too restrictive in some cases
  - downgrading
  - alternative?
- [Volpano and Smith, 2000] “Verifying Secrets and Relative Secrecy”
  - testing secret against a constant leaks the secret but slowly
  - exponential time in length of the secret, assuming uniform distribution
- Extends authors’ previous work [ENTCS 59 No. 3 (2002)]

Example

```
if (H == L)
    X = 0;
else
    X = 1;
```

- "password checking": leaks value of `H` (sometimes!)
- anticipating definitions:
  - leaks at most 1 bit
  - leaks much less than 1 bit if `H` close to uniformly distributed

Example

```
Y = Integer.MIN_VALUE;
while (Y != H) ++Y;
```

- leaks value of `H` every time
- but very slowly (if `H` close to uniform) [Volpano and Smith]
  - not yet captured by our analysis
  - only deal with absolute leakage so far

Quantified leakage

- Idea: use Information Theory (Shannon): how much interference?
- Not new: [Cohen, 1977], [Denning, 1982], [Millen, 1987], [Gray, 1991], . . .
  - non-interference implies 0 bits leaked
- Our contribution:
  - analyse for quantity of information leaked

Information

- Surprise of an event s_i occurring with probability p_i:

$$\log \frac{1}{p_i}$$

- Information (aka entropy) $=$ expected value of surprise:

$$\mathcal{H} \stackrel{\text{def}}{=} \sum_{i=1}^{n} p_i \log \frac{1}{p_i}$$

  - maximised by uniform distribution: $\mathcal{H} = \log n$

Random variables

- Random variable = function from sample space to observation space

$$X : \{s_1, \ldots, s_n\} \to O$$

$$P(X = x) \stackrel{\text{def}}{=} \sum_{s_i \in X^{-1}(x)} p_i$$

- Joint random variable:

$$(X, Y) \stackrel{\text{def}}{=} <X, Y>$$

Conditioning/Restriction

- Random variable X conditioned on $Y = y$:

$$P(X = x|Y = y) \stackrel{\text{def}}{=} \sum_{s_i \in <X,Y>^{-1}(x,y)} p_i / P(Y = y)$$

- Note $(X|Y = y)$ is $X \restriction Y^{-1}(y)$ (the restriction of X to $Y^{-1}(y)$) with distribution normalised in domain

Entropy and conditional entropy

- Expected value of surprise when X is observed:

$$\mathcal{H}(X) \stackrel{\text{def}}{=} \sum_x P(X=x) \log \frac{1}{P(X=x)}$$

- Expected value of entropy of X when Y is known:

$$\mathcal{H}(X|Y) \stackrel{\text{def}}{=} \sum_y P(Y=y)\mathcal{H}(X|Y=y)$$

Joint information and mutual information

- Joint entropy:

$$\mathcal{H}(X, Y) \stackrel{\text{def}}{=} \mathcal{H}(<X, Y>)$$

- Mutual entropy:

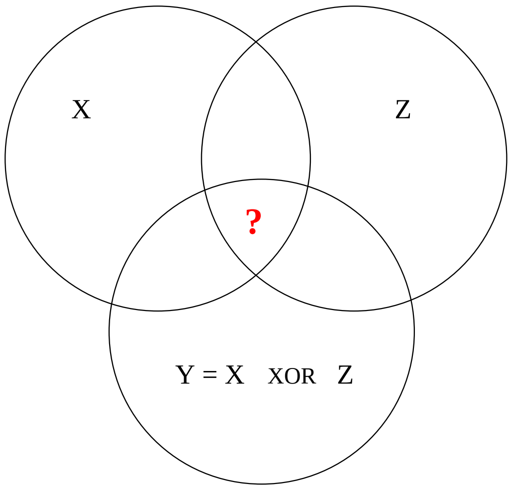
$$\mathcal{I}(X; Y) \stackrel{\text{def}}{=} \mathcal{H}(X) + \mathcal{H}(Y) - \mathcal{H}(X, Y)$$

- Conditional mutual entropy:

$$\mathcal{I}(X; Y|Z) \stackrel{\text{def}}{=} \mathcal{H}(X|Z) + \mathcal{H}(Y|Z) - \mathcal{H}(X, Y|Z)$$

Information is funny stuff!

- $\mathcal{H}(X, Y) \sim [X] \cup [Y]$ $\qquad$ $\mathcal{I}(X;Y) \sim [X] \cap [Y]$
- $\mathcal{H}(X|Y) \sim [X] - [Y]$ $\qquad$ $\mathcal{I}(X;Y|Z) \sim ([X] \cap [Y]) - [Z]$
- BUT: $\mathcal{I}(X;Y) = 0 \not\Rightarrow \mathcal{I}(X;Y|Z) = 0$



Program variables as random variables

- Assume a program in a primitive imperative language (While) with program variables V
  - stores $\sigma \in \Sigma \stackrel{\text{def}}{=} V \rightarrow \{-2^{k-1}, \ldots, 2^{k-1} - 1\}$

- Assume a probability distribution on $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$ (the inputs)
  - random variable X^{in} : the value of X in the input store
  - random variable X^{out}: the value of X when(!) the program terminates

Quantifying leakage/interference

- For simplicity, assume all variables initialised to 0 except H and L
- The quantity of information leaked from H to X:

$$\begin{array}{ll} (1) & \mathcal{L}(X) \stackrel{\text{def}}{=} \mathcal{I}(H^{\text{in}}; X^{\text{out}} | L^{\text{in}}) \\ (2) & \mathcal{L}(X) \stackrel{\text{def}}{=} \mathcal{H}(X^{\text{out}} | L^{\text{in}}) \end{array}$$

  - (1) agrees with [Gray,1991]
  - (1) and (2) equivalent for a deterministic language

Calculating bounds on $\mathcal{L}(X)$

- Want to calculate bounds on $\mathcal{L}(X)$ given bounds on the initial information contained in H:
  - $\mathcal{H}(H^{\mathrm{in}}|L^{\mathrm{in}})$ (the 'real' size of the secret)
- a priori bounds on size of secret: $0 \leq \mathcal{H}(H^{\mathrm{in}}|L^{\mathrm{in}}) \leq k$
  - Can get better results if we know better bounds

Analyse for worst-case choice of L^{in}

- For random variable X, let $X_\lambda \stackrel{\text{def}}{=} (X|L^{\text{in}} = \lambda)$

- Our analysis calculates bounds on $\mathcal{H}(X_\lambda^{\text{out}})$ given bounds on $\mathcal{H}(H_\lambda^{\text{in}})$
  - We calculate/require bounds which hold for all λ
  - Note $\mathcal{H}(H_\lambda^{\text{in}}) = \mathcal{H}(H^{\text{in}}|L^{\text{in}})$ if H^{in} and L^{in} are independent

- Proposition:

$$(\forall \lambda . a \leq \mathcal{H}(X_\lambda^{\text{out}}) \leq b) \Rightarrow a \leq \mathcal{H}(X^{\text{out}}|L^{\text{in}}) \leq b$$

Data Processing theorem

- If $X \to Y \to Z$ then $\mathcal{I}(Z;X) \leq \mathcal{I}(Y;X)$
  - Corollary: if $(\exists f. Z = f(X))$ then $\mathcal{H}(Z) \leq \mathcal{H}(X)$
- Use this as the basis of a 'compositional' analysis:
  1. associate random variables X^n with all program points n (not just in, out)
  2. find $n_1, \ldots, n_j$ such that $\exists f. X^n = f(Y^{n_1}, \ldots, Y^{n_j})$
  3. calculate bounds $a_i \leq \mathcal{H}(Y^{n_i}) \leq b_i$
- DP theorem gives: $\mathcal{H}(X^n) \leq b_1 + \cdots + b_j$
  - But for lower bounds we need to know f. . .

Example

```
1: X = A;
2: if (B) then
3:      Y = C;
   else
4:      X = D;
5: Z = X + 1;
6: ...
```

- $\exists f. Z^6 = f(A^1, B^2, D^4)$

Loops

- Associating random variables to arbitrary program points is not quite straightforward:
  - For each n, need a value for X^n for each choice of input
  - But control may pass through n many times, or not at all

- For a given input, define X^n as the value taken by X at n the final time control passes through n
  - A partial function

- For n inside a loop we need to find dependencies outside the loop
  - In general this will mean we know f exists but we won't know what it is

Beyond the Data Processing theorem

Examples when f is known (assume twos-complement arithmetic):

*****: $a \leq \mathcal{H}(Y) \Rightarrow a \leq \mathcal{H}(Y * n)$, if n is odd

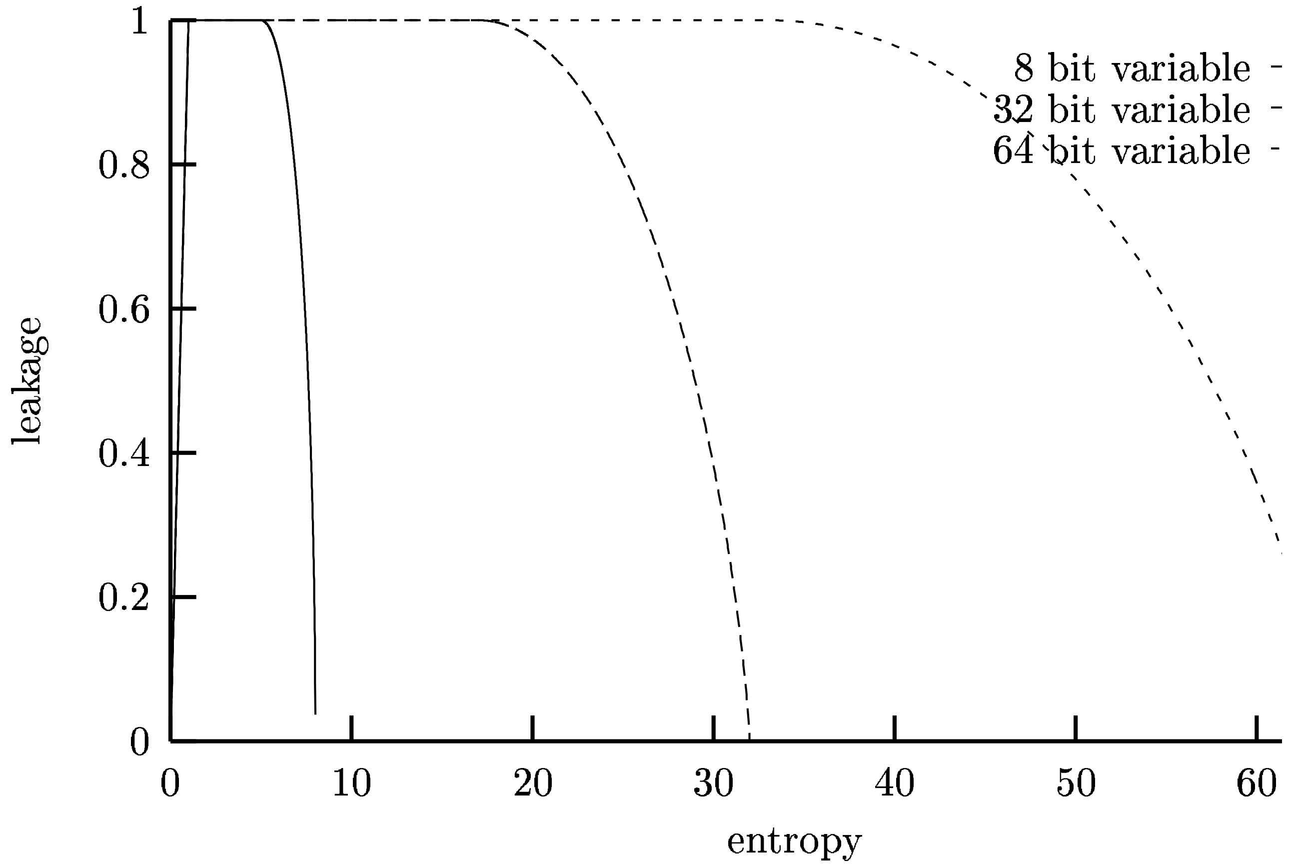
+: $a \leq \mathcal{H}(Y) \wedge \mathcal{H}(Z) \leq b \Rightarrow a - b \leq \mathcal{H}(Y + Z)$

==: $a \leq \mathcal{H}(Y) \wedge \mathcal{H}(Z) \leq b \Rightarrow \mathcal{H}(Y{=}{=}Z) \leq \mathcal{B}(q)$, where:

$$\mathcal{B}(q) \stackrel{\text{def}}{=} q \log \tfrac{1}{q} + (1-q) \log \tfrac{1}{1-q}$$
$$q \leq 0.5$$
$$a - b = \mathcal{U}_k(q) \stackrel{\text{def}}{=} q \log \tfrac{1}{q} + (1-q) \log \tfrac{2^k - 1}{1-q}$$

Note role of k

Verifying secrets



Example

```
in: B = (H==L);
 1: if (B) then
 2:     X = 0;
    else
 3:     X = 1;
 4: Y = 0;
 5: while (X != 0) do
    {
 6:     Y = Y*3;
 7:     X = X-1;
    }
 8: ...
```

- Dependencies:

$$\exists f. Y^8 = f(Y^6)$$
$$\exists g. Y^6 = g(B^1)$$
$$B^1 = (H^{\mathrm{in}}{==}L^{\mathrm{in}})$$

- If $32 = k \leq \mathcal{H}(H^{\mathrm{in}})$:

$$\mathcal{H}(Y^8) \leq 7.8 \times 10^{-9}$$

Further work

- Combine with other static analyses
- Combine with theorem proving
- Richer languages
  - Work in progress on PCF (using Generalised Flowcharts [Malacaria and Hankin, 1998])
- Timing (1): analyse for rates of leakage
- Timing (2): analyse for timing leaks