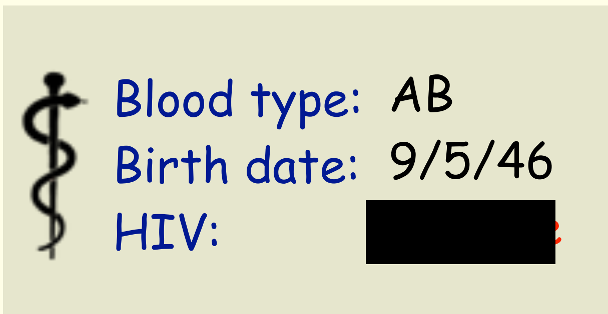
Recent Developments in Quantitative Information Flow

Geoffrey Smith
Florida International University

LICS Tutorial, 9 July 2015

Confidentiality

- Protecting the confidentiality of private information is a fundamental issue in computer security.



- **Access control** and **encryption** are valuable tools, but they cannot stop **authorized** systems from leaking their secret inputs, maliciously or accidentally, to their observable outputs.
- We need to control the flow of information in systems.

Information flow

- Noninterference [Cohen77, GoguenMeseguer82] requires that a system's observable output be **independent** of its secret input.
- Noninterference can be guaranteed by means of **type systems**.
- Unfortunately, noninterference is too strong, because some leakage of sensitive information is often unavoidable in practice.

Motivating example: Statistical database query

- **Secret input**: database of confidential entries
- **Observable output**: result of statistical query (e.g. percentage of population with some disease), possibly with noise added

Motivating example: Password checker

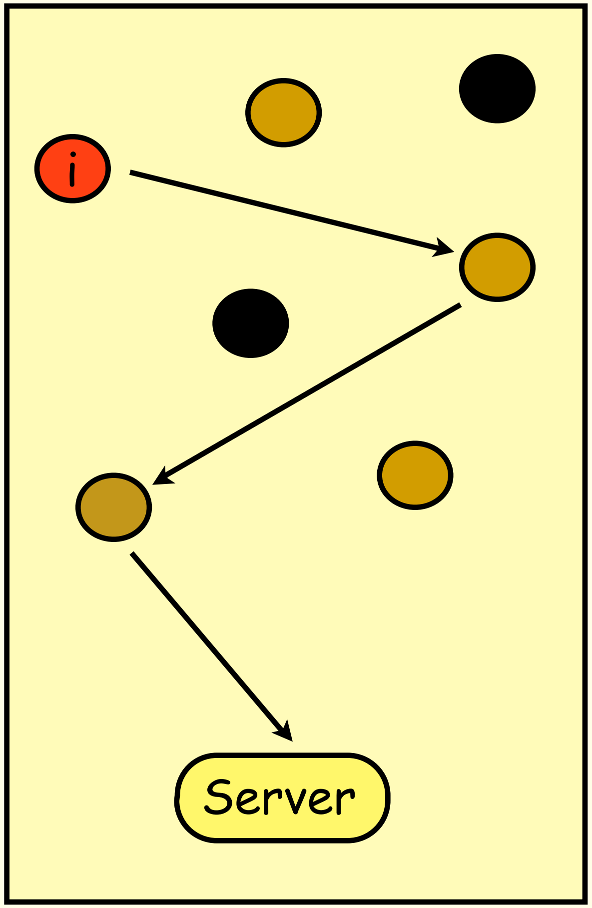
- Check whether guess is equal to password:

```
result = true;
for (i=0; i < N; i++) {
    if (password[i] != guess[i]) {
        result = false;
        break;
    }
}
```

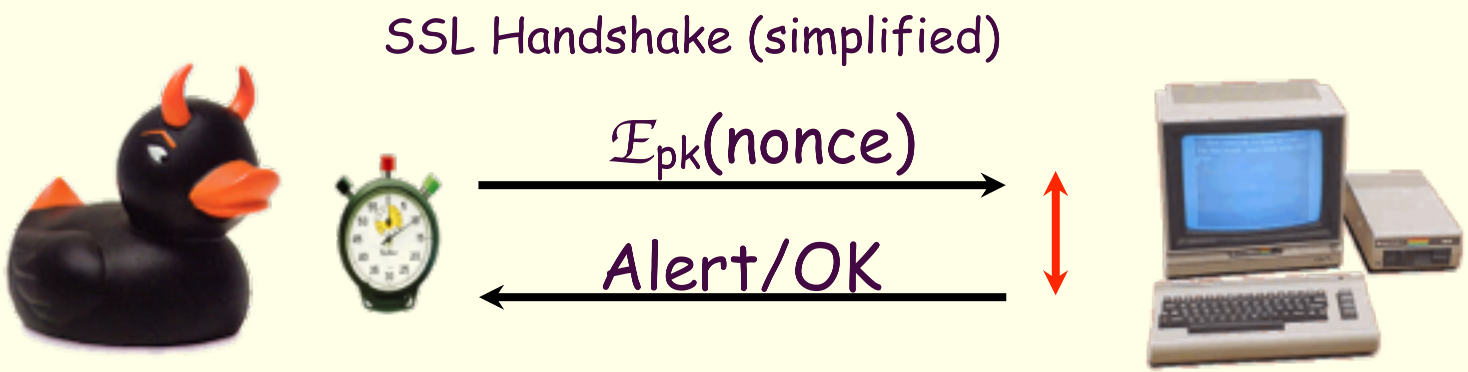
- **Secret input**: password
- **Observable output**:
  - result
  - running time?
    - (If so, the length of the correct prefix is also leaked!)

Motivating example: Crowds protocol [ReiterRubin98]

- Crowd members wish to communicate anonymously with a server.
- The initiator first sends the message to a randomly-chosen forwarder (possibly itself).
- Each forwarder forwards it again with probability p_f, or sends it to the server with probability $1-p_f$.
- But some crowd members are **collaborators** that report who sends them a message.
- **Secret input**: identity of initiator
- **Observable output**: first sender of a message to a collaborator (or no one)



Motivating example: Timing attack on cryptography [BonehBrumley03]



- 1024-bit RSA private key recovered in **2 hours** from standard OpenSSL implementation across LAN.
- **Secret input**: RSA private key
- **Observable output**: approximate timings of decryptions of a sequence of nonces

Plan of the talk

- Motivation
- Concepts of Quantitative Information Flow
  - Channels, hyper-distributions, vulnerability, min-entropy leakage, g-leakage
- Robustness
  - Robust channel ordering: composition refinement
  - Capacity: multiplicative and additive
- Conclusion

Information-theoretic channels

- The earlier examples can all be modeled as **channels**.
- Finite sets $\mathcal{X}$ (secret inputs), $\mathcal{Y}$ (observable outputs).
  - The choice of $\mathcal{Y}$ is subtle, and crucial!
- On input X, the channel probabilistically outputs Y.
- Channel matrix *C* gives the conditional probabilities p(y|x):

	y_1	y_2	y_3	y_4
x_1	1/2	1/2	0	0
x_2	0	1/4	1/2	1/4
x_3	1/2	1/3	1/6	0

- The rows of *C* sum to 1.
- *C* is **deterministic** if each row contains exactly one 1.

Quantitative Information Flow

- We wish to quantify the leakage of secret input X to observable output Y caused by channel C, allowing us to argue that some leaks are "small".
- The secrecy of X is modeled by a prior distribution π on $\mathcal{X}$.
- Both π and C are assumed known by the adversary $\mathcal{A}$.
- **Key insight:**
  The (information-theoretic) essence of C is a mapping from priors π to hyper-distributions $[\pi,C]$, which are distributions on posterior distributions.

Example

Prior

π
1/4
1/2
1/4

Channel matrix

C	y_1	y_2	y_3	y_4
x_1	1/2	1/2	0	0
x_2	0	1/4	1/2	1/4
x_3	1/2	1/3	1/6	0

Example

Prior

π
1/4
1/2
1/4

Channel matrix

C	y_1	y_2	y_3	y_4
x_1	1/2	1/2	0	0
x_2	0	1/4	1/2	1/4
x_3	1/2	1/3	1/6	0

Scale rows with π.

Joint matrix

J	y_1	y_2	y_3	y_4
x_1	1/8	1/8	0	0
x_2	0	1/8	1/4	1/8
x_3	1/8	1/12	1/24	0

Example

Prior

π
1/4
1/2
1/4

Channel matrix

C	y_1	y_2	y_3	y_4
x_1	1/2	1/2	0	0
x_2	0	1/4	1/2	1/4
x_3	1/2	1/3	1/6	0

Scale rows with π.

Joint matrix

J	y_1	y_2	y_3	y_4
x_1	1/8	1/8	0	0
x_2	0	1/8	1/4	1/8
x_3	1/8	1/12	1/24	0

Add up columns.

Distribution on Y

p_Y	1/4	1/3	7/24	1/8

Example

Prior

π
1/4
1/2
1/4

Channel matrix

C	y_1	y_2	y_3	y_4
x_1	1/2	1/2	0	0
x_2	0	1/4	1/2	1/4
x_3	1/2	1/3	1/6	0

Scale rows with π.

Joint matrix

J	y_1	y_2	y_3	y_4
x_1	1/8	1/8	0	0
x_2	0	1/8	1/4	1/8
x_3	1/8	1/12	1/24	0

Add up columns.

Distribution on Y

p_Y	1/4	1/3	7/24	1/8

Normalize columns.

Posterior distributions

	$p_{X\|y1}$	$p_{X\|y2}$	$p_{X\|y3}$	$p_{X\|y4}$
x_1	1/2	3/8	0	0
x_2	0	3/8	6/7	1
x_3	1/2	1/4	1/7	0

Example

Prior

π
1/4
1/2
1/4

Channel matrix

C	y_1	y_2	y_3	y_4
x_1	1/2	1/2	0	0
x_2	0	1/4	1/2	1/4
x_3	1/2	1/3	1/6	0

Scale rows with π.

Joint matrix

J	y_1	y_2	y_3	y_4
x_1	1/8	1/8	0	0
x_2	0	1/8	1/4	1/8
x_3	1/8	1/12	1/24	0

Add up columns.

Distribution on Y

p_Y	1/4	1/3	7/24	1/8

Normalize columns.

Posterior distributions

	$p_{X\|y_1}$	$p_{X\|y_2}$	$p_{X\|y_3}$	$p_{X\|y_4}$
x_1	1/2	3/8	0	0
x_2	0	3/8	6/7	1
x_3	1/2	1/4	1/7	0

Drop output labels.

Hyper-distribution on X

$[\pi,C]$	1/4	1/3	7/24	1/8
x_1	1/2	3/8	0	0
x_2	0	3/8	6/7	1
x_3	1/2	1/4	1/7	0

Example

Prior

π
1/4
1/2
1/4

Channel matrix

C	y_1	y_2	y_3	y_4
x_1	1/2	1/2	0	0
x_2	0	1/4	1/2	1/4
x_3	1/2	1/3	1/6	0

Scale rows with π.

Joint matrix

J	y_1	y_2	y_3	y_4
x_1	1/8	1/8	0	0
x_2	0	1/8	1/4	1/8
x_3	1/8	1/12	1/24	0

Add up columns.

Distribution on Y

p_Y	1/4	1/3	7/24	1/8

Normalize columns.

Posterior distributions

	$p_{X\|y_1}$	$p_{X\|y_2}$	$p_{X\|y_3}$	$p_{X\|y_4}$
x_1	1/2	3/8	0	0
x_2	0	3/8	6/7	1
x_3	1/2	1/4	1/7	0

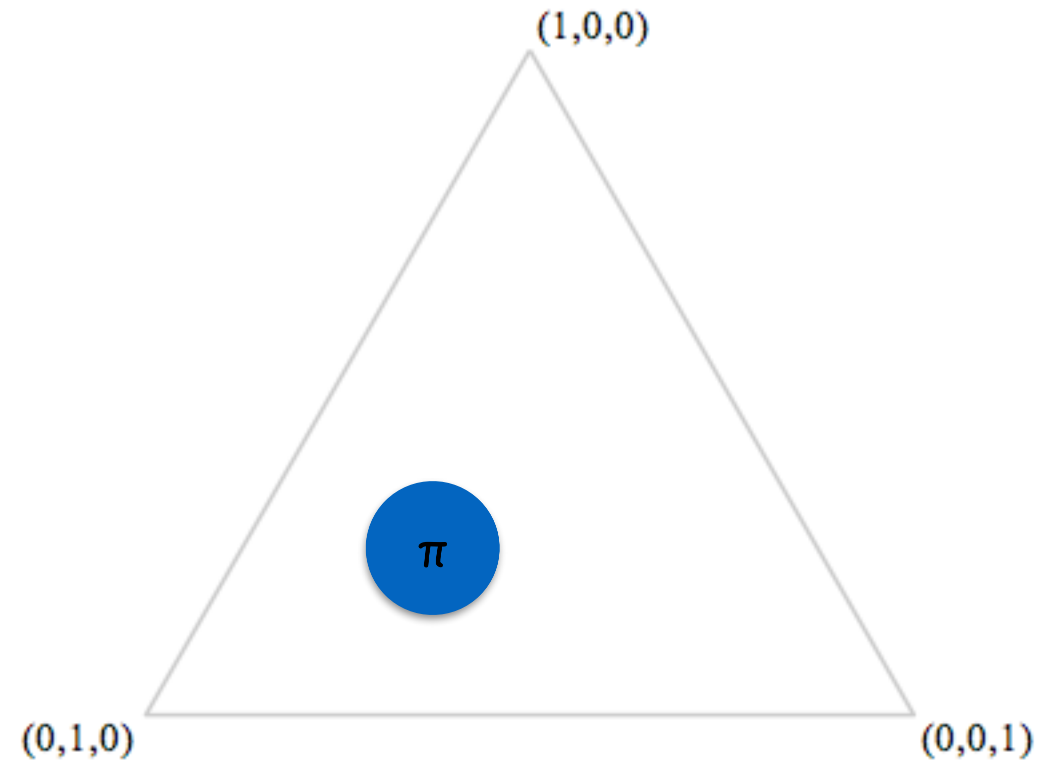
Drop output labels.

Abstractly, a channel is a mapping from priors to hyper-distributions [McIverMeinickeMorgan10].

Hyper-distribution on X

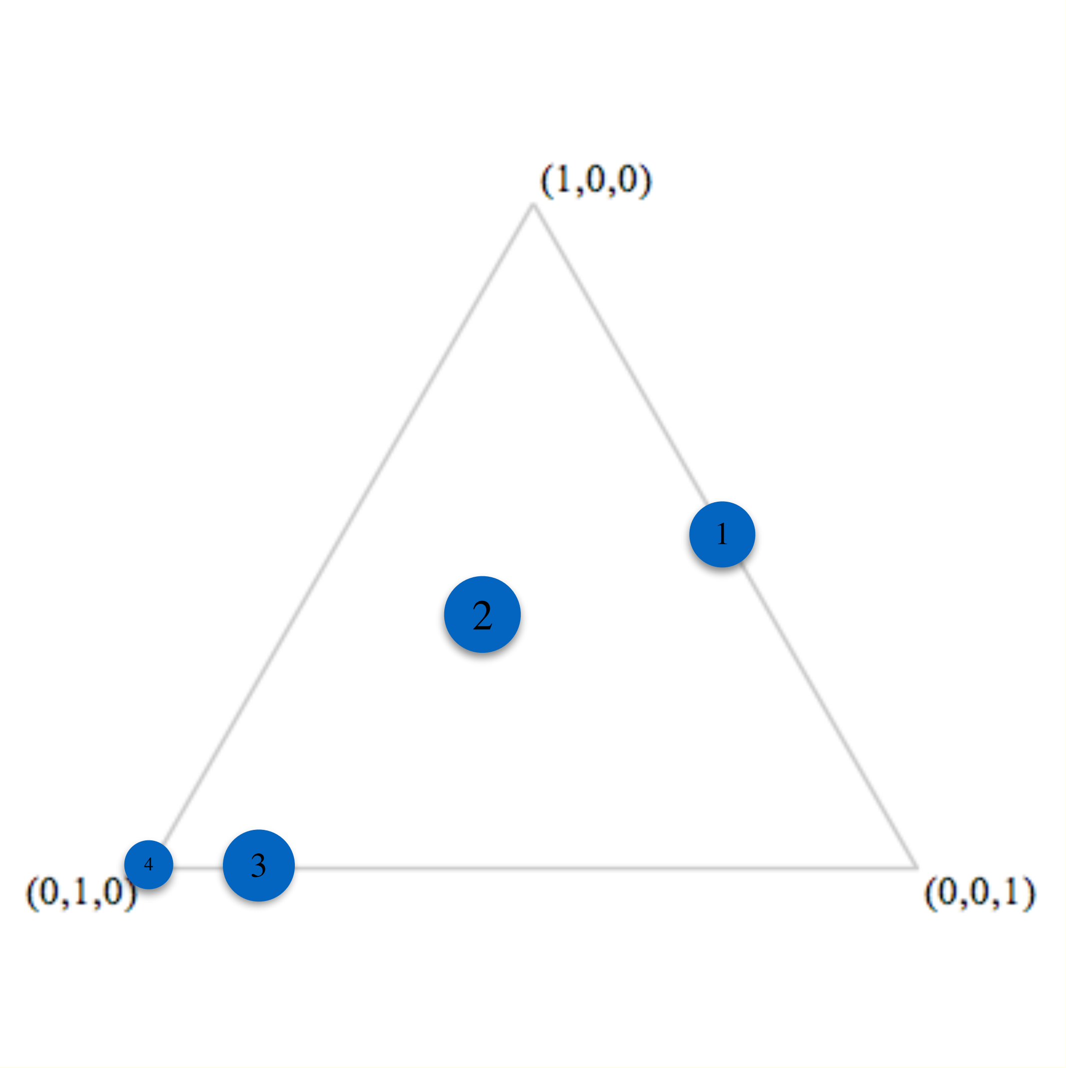
$[\pi,C]$	1/4	1/3	7/24	1/8
x_1	1/2	3/8	0	0
x_2	0	3/8	6/7	1
x_3	1/2	1/4	1/7	0

Graphical representation of example



$\pi = (1/4, 1/2, 1/4)$

Graphical representation of example



Leakage = mutual information $I(X;Y)$?

- It is tempting to measure leakage using Shannon entropy and mutual information:
  - $H(X) = -\Sigma_x \pi_x \log \pi_x$ "prior uncertainty"
  - $H(X|Y) = \Sigma y\ p(y)\ H[p_{X|y}]$ "posterior uncertainty"
  - $I(X;Y) = H(X) - H(X|Y)$ "leakage"
- But consider the channel

  $$Y = X \quad {}_{\frac{1}{8}}{+} \quad Y = -1$$

- If X is a uniformly-distributed 64-bit unsigned int, we get $H(X) = 64$, $H(X|Y) = 56$, and $I(X;Y) = 8$ bits.
- We might expect that $H(X|Y) = 56$ means that the secrecy of X has not been harmed much.
- But $\frac{1}{8}$ of the time the adversary learns X exactly!

Vulnerability [Smith09]

- For confidentiality, it seems more useful to measure leakage based on X's vulnerability to be guessed correctly by $\mathcal{A}$ in one try.
- Prior vulnerability:
  $V[\pi] = \max_x \pi_x$
- Posterior vulnerability:
  $V[\pi,C] = \sum_y p(y)\, V[p_{X|y}]$
  - $V[\pi,C]$ is the average vulnerability in the hyper-distribution.
  - $V[\pi,C]$ is the complement of the Bayes risk.

Operational significance of vulnerability

- $V[\pi]$ is an optimal adversary $\mathcal{A}$'s probability of winning the following game:

$$x \xleftarrow{\$} \pi$$
$$w \xleftarrow{\$} \mathcal{A}(\pi)$$
if $w = x$ then **win** else **lose**

- $V[\pi, C]$ is an optimal adversary $\mathcal{A}$'s probability of winning the following game:

$$x \xleftarrow{\$} \pi$$
$$y \xleftarrow{\$} C_{x,-}$$
$$w \xleftarrow{\$} \mathcal{A}(\pi, C, y)$$
if $w = x$ then **win** else **lose**

Min-entropy leakage

- Min-entropy leakage is defined multiplicatively:

$$\mathcal{L}(\pi,C) = \log \frac{V[\pi,C]}{V[\pi]}$$

- Note: -log V[π] is Rényi's min-entropy $H_\infty[\pi]$.

- Later, we will also consider additive leakage:

$$\mathcal{L}^{+}(\pi,C) = V[\pi,C] - V[\pi]$$

A surprising example ("base-rate fallacy")

- Consider a good, but imperfect, test for cancer:

C	positive	negative
cancer	9/10	1/10
no cancer	1/10	9/10

- Prior (age 40-50, no symptoms, no family history)
  $\pi[\text{cancer}] = 1/100$ $\pi[\text{no cancer}] = 99/100$

A surprising example ("base-rate fallacy")

- Consider a good, but imperfect, test for cancer:

C	positive	negative
cancer	9/10	1/10
no cancer	1/10	9/10

- Prior (age 40-50, no symptoms, no family history)
  $\pi[\text{cancer}] = 1/100$ $\pi[\text{no cancer}] = 99/100$

$[\pi, C]$	27/250	223/250
cancer	1/12	1/892
no cancer	11/12	891/892

- $\mathcal{A}$'s best guess is always guess "no cancer"!
- $V[\pi, C] = 0.99 = V[\pi]$, so $\mathcal{L}(\pi, C) = 0$.

Limitations of min-entropy leakage

- Vulnerability V has a clear operational significance, but it implicitly assumes that adversary $\mathcal{A}$ benefits only by guessing X exactly and in one try.
- But many other scenarios are possible:
  - Maybe $\mathcal{A}$ can benefit by guessing X partially or approximately.
  - Maybe $\mathcal{A}$ is allowed to make multiple guesses.
  - Maybe $\mathcal{A}$ is penalized for making a wrong guess.
- No single leakage measure is appropriate in all scenarios.

Generalizing to g-vulnerability and g-leakage [AlvimChatzikokolakisPalamidessiSmith12]

- We can model each scenario with a **gain function** g.
  - Finite set $\mathcal{W}$ of guesses (or "actions") about X.
  - $g : \mathcal{W} \times \mathcal{X} \to [0, 1]$
  - $g(w,x)$ gives the value of w if the secret is x.
  - Note: Ordinary vulnerability implicitly uses
    $g_{id}(w,x) = \begin{cases} 1, \text{ if } w = x \\ 0, \text{ otherwise} \end{cases}$
- **Prior g-vulnerability**: $V_g[\pi] = \max_w \sum_x \pi_x \, g(w,x)$
  - maximum expected gain over all possible guesses
- **Posterior g-vulnerability**: $V_g[\pi,C] = \sum_y p(y) \, V_g[p_{X|y}]$
- **g-leakage**: $\mathcal{L}_g(\pi,C) = \log \left(V_g[\pi,C]/V_g[\pi]\right)$

Plan of the talk

- Motivation
- Concepts of Quantitative Information Flow
  - Channels, hyper-distributions, vulnerability, min-entropy leakage, g-leakage
- Robustness
  - Robust channel ordering: composition refinement
  - Capacity: multiplicative and additive
- Conclusion

Robustness worries

- Using g-leakage, we can express precisely a rich variety of operational scenarios.
- But we could worry about the **robustness** of our conclusions about leakage.
- The g-leakage $\mathcal{L}_g(\pi,C)$ depends on both π and g.
  - π models adversary $\mathcal{A}$'s prior knowledge about X
  - g models (among other things) what is valuable to $\mathcal{A}$.
- How confident can we be about these?
- Can we minimize sensitivity to questionable assumptions about π and g?

Channel ordering

- Given channels C and D on secret input X, the question of which leaks more will ordinarily depend on the prior and gain function used.
- Example: (assume 64-bit, uniform unsigned X)

  C. $Y = X \;\; {}_{\frac{1}{8}}\!\!+ \;\; Y = -1$

  D. Y = X | 0x7;
- Both have min-entropy leakage of 61.0 bits.
- We can distinguish them with gain functions.
- g_3, which allows 3 tries, makes D leak more than C.
- g_{tiger}, which gives a penalty for a wrong guess (allowing "⊥" for "don't guess") makes C leak more.

Robust channel ordering

- Is there a **robust** ordering?
  - This could support a stepwise refinement methodology.
- Yes!

- **Def**: C is composition refined by D, written $C \sqsubseteq_o D$,
  if D = CE, for some channel E.
  - CE is the cascade of C and E, formed by multiplying the channel matrices C and E.
  - Intuitively, the adversary should never prefer D to C, since he could do the "post-processing" E himself.

Composition refinement and leakage

- **Theorem** [MMSEM14]:
  C is composition refined by D
  iff
  D never leaks more than C, regardless of π and g.

- The forward direction is a generalized data-processing inequality.
- The backward ("coriaceous") direction uses the separating hyperplane lemma to construct the g needed in the contrapositive.
- We later learned that this theorem was proved in 1953 by statistician David Blackwell!

Structure of channels under composition refinement

- Composition refinement is only a pre-order on channel matrices.
- But channel matrices contain **redundant structure** with respect to their abstract denotation as mappings from priors to hyper-distributions.

C	y_1	y_2	y_3
x_1	1	0	0
x_2	1/4	1/2	1/4
x_3	1/2	1/3	1/6

D	z_1	z_2	z_3
x_1	2/5	0	3/5
x_2	1/10	3/4	3/20
x_3	1/5	1/2	3/10

C and D composition refine each other, but they are actually the same abstract channel!

- **Theorem:** On abstract channels, composition refinement. is a partial order.

Capacity

- Another approach to robustness is to consider capacity, the **maximum** leakage of a channel C over **all** priors π and/or gain functions g.
- This gives **worst-case** bounds on leakage.
- There are six capacity scenarios:
  - multiplicative or additive leakage
  - maximize over π, over g, or over **both** π and g.

- I will discuss just a few of these cases.
- A number of them are not well understood yet.

Multiplicative capacity

- Fixing g to g_{id} (giving ordinary vulnerability) and maximizing over π gives interesting results.
- **Def**: Min-capacity $\mathcal{ML}(C) = \sup_\pi \log(V[\pi,C]/V[\pi])$
- **Theorem**: $\mathcal{ML}(C)$ is the log of the sum of the column maximums of C. It is realized on a uniform prior π.
  - **Corollary**: $\mathcal{ML}(C) = 0$ iff the rows of C are identical.
  - **Corollary**: For deterministic C, $\mathcal{ML}(C)$ is the log of the number of feasible output values.

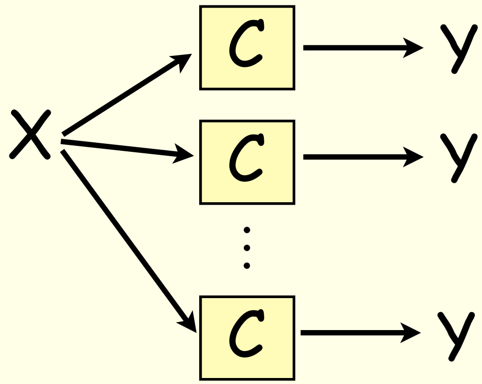
Multiplicative capacity

- Fixing g to g_{id} (giving ordinary vulnerability) and maximizing over π gives interesting results.
- **Def**: Min-capacity $\mathcal{ML}(C) = \sup_{\pi} \log(V[\pi,C]/V[\pi])$
- **Theorem**: $\mathcal{ML}(C)$ is the log of the sum of the column maximums of C. It is realized on a uniform prior π.
  - **Corollary**: $\mathcal{ML}(C) = 0$ iff the rows of C are identical.
  - **Corollary**: For deterministic C, $\mathcal{ML}(C)$ is the log of the number of feasible output values.
- **Theorem** ("Miracle"): Min-capacity is an upper bound on g-leakage, for **every** prior π and gain function g: $\mathcal{ML}(C) \geq \mathcal{L}_g(\pi,C)$.

Example

- Let the secret be an array X containing 10-bit, uniformly distributed passwords for 1000 users.
- Let C be
  $u \xleftarrow{\$} \{0..999\};\ Y = (u, X[u]);$
- $V[\pi] = 2^{-10000}$ and $V[\pi,C] = 2^{-9990}$, so $\mathcal{ML}(C) = 10$
- Now specify that $\mathcal{A}$ gains by guessing **any** user's password:
  - $W = \{ (u,x) \mid 0 \le u \le 999 \text{ and } 0 \le x \le 1023 \}$
  - $g((u,x), X) = \begin{cases} 1, & \text{if } X[u] = x \\ 0, & \text{otherwise} \end{cases}$
  - $V_g[\pi] = 2^{-10}$ and $V_g[\pi,C] = 1$, so $\mathcal{L}_g(\pi,C) = 10$
  - (The Miracle Theorem bounds $\mathcal{L}_g(\pi,C)$, not $V_g[\pi,C]$.)

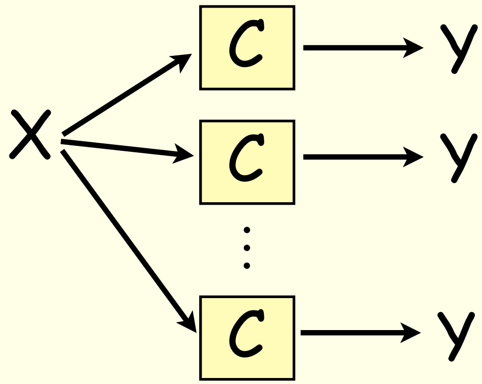
Repeated independent runs $C^{(n)}$



(Useful only when
C is probabilistic!)

- **Theorem**: $\mathcal{ML}(C^{(n)}) \leq |\mathcal{Y}| \log(n+1)$.

Repeated independent runs $C^{(n)}$



(Useful only when
C is probabilistic!)

- **Theorem**: $\mathcal{ML}(C^{(n)}) \leq |\mathcal{Y}| \log(n+1)$.
- Application to timing attacks on **blinded** cryptography:
  - **Blinding** randomizes the ciphertext before decryption, and de-randomizes after decryption.
  - As a result, the n-observation timing attack is a repeated independent runs channel $C^{(n)}$.
  - Hence its min-capacity grows only logarithmically in n.

Additive capacity [ACMMPS14]

- **Theorem**: With respect to g_{id}, the additive capacity of C over all priors π is NP-complete.
  - Notice that the input here is the **channel matrix** C, rather than a concise program.

- **Theorem**: If we fix π and universally quantify over g (ranging over "1-spanning" gain functions), then the additive capacity of C is the Kantorovich distance between the prior $[\pi]$ and hyper-distribution $[\pi,C]$.
  - Hence it is the earth-moving distance between $[\pi]$ and $[\pi,C]$, which can be computed in time linear in $|C|$.

Example

π
1/4
1/2
1/4

C	y_1	y_2	y_3	y_4
x_1	1/2	1/2	0	0
x_2	0	1/4	1/2	1/4
x_3	1/2	1/3	1/6	0

$[\pi,C]$	1/4	1/3	7/24	1/8
x_1	1/2	3/8	0	0
x_2	0	3/8	6/7	1
x_3	1/2	1/4	1/7	0

- If we want the additive capacity over all g, we take the average earth-moving distance between π and each of the posterior distributions.
  - E.g. the distance between π and (1/2, 0, 1/2) is 1/2.
- Overall we get $1/4 \cdot 1/2 + 1/3 \cdot 1/8 + 7/24 \cdot 5/14 + 1/8 \cdot 1/2 = 1/3$

Plan of the talk

- Motivation
- Concepts of Quantitative Information Flow
  - Channels, hyper-distributions, vulnerability, min-entropy leakage, g-leakage
- Robustness
  - Robust channel ordering: composition refinement
  - Capacity: multiplicative and additive
- Conclusion

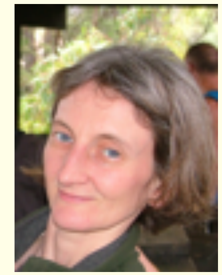
Conclusion

- **Min-entropy** and **g-leakage** allows the quantification of leakage with strong operational significance for confidentiality.
- **Composition refinement** and **capacity** support robust conclusions about leakage.
- Research directions:
  - Static analysis of leakage in programs
  - Relationship with differential privacy
  - Computational measures of leakage
  - Generalizing from channels to Hidden Markov Models

Many thanks to my collaborators


Catuscia Palamidessi


Kostas Chatzikokolakis


Annabelle McIver


Carroll Morgan


Boris Köpf

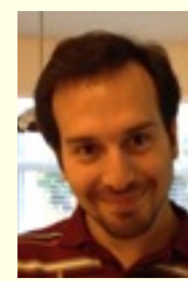

Miguel Andrés


Mário Alvim


Ziyuan Meng


Barbara Espinoza


Marco Stronati


Nico Bordenabe

Questions?