

Quantifying Information Flow Using Min-Entropy

Geoffrey Smith
Florida International University

Royal Holloway ISG Research Seminar, 13 October 2011

1

Secure Information Flow

- Protecting the **confidentiality** of secret information is a fundamental issue in computer security:

Blood type: AB
Birth date: 9/5/46
HIV: ██████████



- Access control and encryption are not sufficient!
- Systems should not allow secret information to leak to their publicly observable outputs.
 - Crucial (and subtle) question: what is publicly observable?**

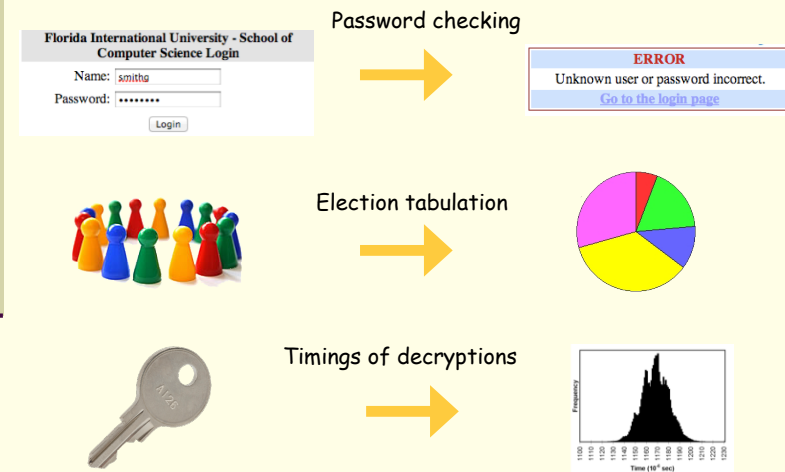
2

The Denning Restrictions and Noninterference

- [DenningDenning77]
 - Let S be a secret input, O a public output.
 - Explicit flow: $O = S/3 + 1$;
 - Implicit flow: **if** ($S \% 2 == 0$) $O = 0$; **else** $O = 1$;
- [VolpanoSmithIrvine96]
 - In deterministic programs, a type system preventing explicit and implicit flows ensures **noninterference**:
 - Running the program with two different initial values of S gives the **same** final value of O (so long as both runs terminate successfully).
 - So the final value of O reveals **no** information about S .
- [Myers, Sabelfeld, Sands, Zdancewic, ...]

3

But some leakage is often unavoidable



4

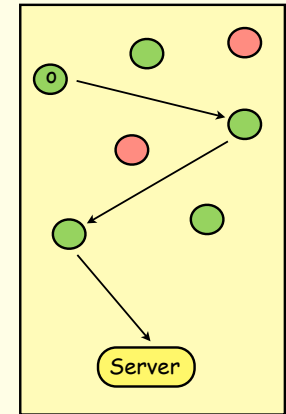
Quantitative information flow

- A **quantitative** theory lets us talk about “how much” information is leaked to an adversary \mathcal{A} who sees the observable output.
- Then “small” leaks may be tolerated.
- This has been an active area of research for the past decade [ClarkHuntMalacaria02, ...]
- A first, straightforward, example: $O = S \& 0777$;
 - If S is a 64-bit integer, and all 2^{64} values are equally likely, then this program leaks 9 bits (out of 64) to O .

5

A more complicated example: Crowds Protocol [RubinReiter98]

- Users wish to communicate anonymously with a server.
- The originator 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.
- Some information about the originator may be leaked. But how much???



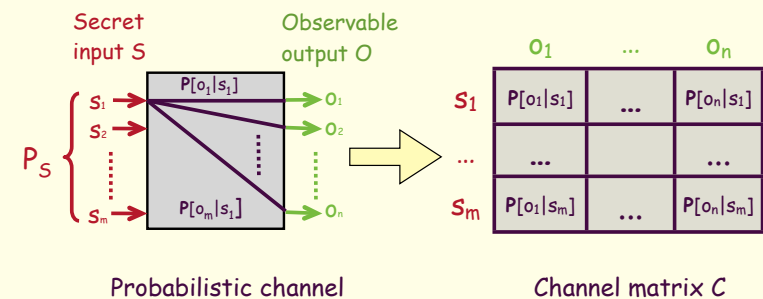
6

Plan of the talk

- Motivation
- **Information-theoretic channels**
- Quantifying leakage
 - using mutual information
 - using min-entropy
 - channel capacity
- Channels in cascade
 - application to timing attacks on cryptography
- Some techniques for calculating min-entropy leakage

7

Information-theoretic channels



Each row of C sums to 1. C is **deterministic** if each entry is 0 or 1.

Random variable S is chosen according to a priori distribution P_S .

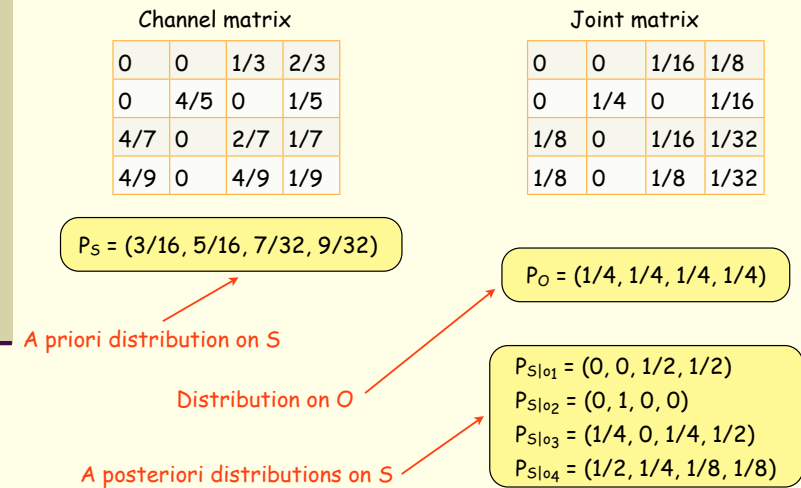
8

Joint and a posteriori distributions

- Multiplying row s of C by $P_S[s]$ gives the **joint matrix** $P[s,o] = P_S[s]C[s,o]$
- By marginalization, we get a random variable O with distribution $P[o] = \sum_s P[s,o]$.
- For each value o of O , we also get an **a posteriori distribution** $P_{S|o}$ by normalizing column o of the joint matrix.
- Assuming that \mathcal{A} knows C and P_S , the distribution $P_{S|o}$ is what \mathcal{A} knows about S if it sees output o .

9

An example channel and its distributions



10

Plan of the talk

- Motivation
- Information-theoretic channels
- **Quantifying leakage**
 - using mutual information
 - using min-entropy
 - channel capacity
- Channels in cascade
 - application to timing attacks on cryptography
- Some techniques for calculating min-entropy leakage

11

Quantifying leakage

- **How much** information about S is leaked to an adversary \mathcal{A} seeing O ?
- Key quantities to define:
 - \mathcal{A} 's initial uncertainty about S
 - \mathcal{A} 's remaining uncertainty about S
 - leakage to O
- Intuitive equation:

"leakage = initial uncertainty - remaining uncertainty"
- Clearly these "uncertainties" depend on the a priori and a posteriori distributions on S .
- But how should they be defined???

12

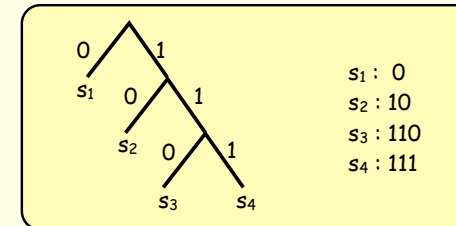
Shannon entropy [1948]

- A classic measure of "uncertainty"
- Let S be a random variable with distribution P_S
- Definition: $H(S) = -\sum_s P_S[s] \log P_S[s]$
- Examples:
 - On a uniform distribution $P_S = (1/n, 1/n, \dots, 1/n)$,
 $H(S) = -n (1/n) \log (1/n) = \log n$
 - If $P_S = (1/2, 1/4, 1/8, 1/8)$,
 $H(S) = (1/2)\log 2 + (1/4)\log 4 + (1/8)\log 8 + (1/8)\log 8$
 $= (1/2)1 + (1/4)2 + (1/8)3 + (1/8)3$
 $= 7/4$

13

Operational significance?

- If $P_S = (1/2, 1/4, 1/8, 1/8)$, we get the following **Huffman code** for values of S :



- Average code length
 $= (1/2)1 + (1/4)2 + (1/8)3 + (1/8)3 = 7/4 = H(S)$
- **Shannon's source coding theorem**: $H(S)$ is the average number of bits required to transmit S .

14

Conditional Shannon entropy

- $H(S)$ is a plausible measure of \mathcal{A} 's initial uncertainty.
- \mathcal{A} 's remaining uncertainty could be defined as the weighted average of the Shannon entropy of the a posteriori distributions $P_{S|o}$.
- Definition: $H(S|O) = \sum_o P[o] H(S|o)$
- This can be seen as the average number of bits required to transmit S , given O .
- On the example channel from before,
 $H(S|O) = (1/4)(1 + 0 + 3/2 + 7/4) = 17/16$

15

Mutual information leakage

- initial uncertainty = $H(S)$
- remaining uncertainty = $H(S|O)$
- leakage = $H(S) - H(S|O)$
- $H(S) - H(S|O)$ is the **mutual information** $I(S;O)$
- [ClarkHuntMalacaria05, ClarksonMyersSchneider05, KöpfBasin07, ChatzikokolakisPalamidessiPanangaden08, ...]
- $I(S;O) \geq 0$.
- And $I(S;O) = 0$ iff S and O are independent.
- If C is **deterministic**, then leakage simplifies to $H(O)$.
 - $I(S;O) = I(O;S) = H(O) - H(O|S) = H(O) - 0 = H(O)$

16

Operational significance?

- For security, the average number of bits required to transmit S **reliably** isn't really the key question.
- Instead, we are more worried about the risk that adversary \mathcal{A} **might** discover the value of S .
- There is a strong bound on the **guessing entropy**, $G(S)$, the expected number of tries required to guess S :
 - Theorem [Massey94]:** $G(S) > (1/4)2^{H(S)}$.
 - Similarly, $G(S|O) > (1/4)2^{H(S|O)}$.
- This is good, but it can be very misleading...
- If $P_S = (1/2, 2^{-1000}, 2^{-1000}, 2^{-1000}, 2^{-1000}, \dots, 2^{-1000})$, then $G(S) \approx 2^{997}$, even though s_1 is correct half the time!

17

Two key examples

- Assume $0 \leq S < 2^{64}$, uniformly distributed.
- if $(S \% 8 == 0) O = S$; else $O = 1$:
 - mutual information leakage $I(S;O) \approx 8.17$
 - remaining uncertainty $H(S|O) \approx 55.83$
 - \mathcal{A} 's expected probability of guessing S in one try, given O , exceeds $1/8$.
- $O = S \ \& \ 0777$:
 - mutual information leakage $I(S;O) = 9$
 - remaining uncertainty $H(S|O) = 55$
 - \mathcal{A} 's expected probability of guessing S in one try, given O , is $1/2^{55}$.

18

Bayes Vulnerability

- Shannon entropy and mutual information thus do not give very satisfactory confidentiality properties.
- So we seek another measure of the "uncertainty" of a probability distribution.
- [Smith09] proposed measuring uncertainty in terms of S 's **vulnerability** to being guessed by \mathcal{A} in one try.
- Definition: $V(S) = \max_s P_S[s]$
- Definition: $V(S|O) = \sum_o P[o] V(S|o)$
- $V(S|O) = \sum_o P[o] \max_s P[s|o] = \sum_o \max_s P[s,o]$
- $V(S|O)$ is the complement of the **Bayes risk**.

19

$V(S)$ and $V(S|O)$ on example channel

Channel matrix				Joint matrix			
0	0	1/3	2/3	0	0	1/16	1/8
0	4/5	0	1/5	0	1/4	0	1/16
4/7	0	2/7	1/7	1/8	0	1/16	1/32
4/9	0	4/9	1/9	1/8	0	1/8	1/32

A priori distribution P_S
(3/16, 5/16, 7/32, 9/32)

- $V(S) = \max_s P_S[s] = 5/16$
- $V(S|O) = \sum_o \max_s P[s,o] = 1/8 + 1/4 + 1/8 + 1/8 = 5/8$
- S 's expected vulnerability doubles.
- A priori, \mathcal{A} guesses that S is s_2 .
- A posteriori, \mathcal{A} 's best guess for S depends on O :
 - $o_1 \rightarrow s_3$ (or s_4), $o_2 \rightarrow s_2$, $o_3 \rightarrow s_4$, $o_4 \rightarrow s_1$

20

Adversary \mathcal{A} 's guessing strategy

- A priori, guess some s that maximizes $P_S[s]$.
- A posteriori, given o , guess some s that maximizes $P[s,o]$.
- Takes time linear in the size of the channel matrix C .
- But suppose C takes as input a 100-digit prime p , and outputs pq , where q is a uniformly-distributed 101-digit prime.
- Then $V(S|O) = 1$, since each column of C has a unique nonzero entry—but it isn't easy for \mathcal{A} to find it!
 - By the prime number theorem, C has over 10^{97} rows.
- Vulnerability is **information theoretic**, not **computational**.

21

Min-entropy leakage

- Convert from vulnerability to uncertainty by taking the negative logarithm.
- This gives **min-entropy** [Rényi61].
- $H_\infty(S) = -\log V(S)$
- $H_\infty(S|O) = -\log V(S|O)$
 - (This definition is not universally agreed-upon...)
- **Min-entropy leakage**
$$\mathcal{L}_{SO} = H_\infty(S) - H_\infty(S|O) = \log \frac{V(S|O)}{V(S)}$$
- So leaking x bits means increasing the expected vulnerability by a factor of 2^x .

22

Min-entropy leakage of key examples

- (Recall: $0 \leq S < 2^{64}$, uniformly distributed)
- **if $(S \% 8 == 0)$ $O = S$; else $O = 1$;**
 - $\mathcal{L}_{SO} \approx 61.00$ $[I(S;O) \approx 8.17]$
 - $H_\infty(S|O) \approx 3.00$ $[H(S|O) \approx 55.83]$
- **$O = S \& 0777$;**
 - $\mathcal{L}_{SO} = 9$ $[Same\ as\ I(S;O)]$
 - $H_\infty(S|O) = 55$ $[Same\ as\ H(S|O)]$

23

One-guess vulnerability?

- Compare
 - (1) **if $(S \% 8 == 0)$ $O = S$; else $O = 1$;**
 - (2) **$O = S | 07$;**
- Both have min-entropy leakage of 61.00 bits?
- (1) reveals almost nothing seven-eighths of the time.
- (2) always reveals all but the last three bits of S .
- But if a wrong guess triggers an alarm, (1) is perhaps worse—whenever $O \neq 1$, \mathcal{A} knows S exactly.
- No single measure is ideal in all scenarios...
- If V_i denotes **i -guess** vulnerability, $V_i(S|O) \leq i V(S|O)$.

24

Properties of min-entropy leakage

- $H(S) \geq H_\infty(S)$
 - $H(S) = H_\infty(S)$ if P_S is uniform
- $H(S|O) \geq H_\infty(S|O)$ [SanthiVardy06]
- So, with a uniform a priori, $I(S;O) \leq \mathcal{L}_{SO}$.
- But, in general, no relation holds:
 - $I(S;O) = 0$ iff S and O are independent.
 - $\mathcal{L}_{SO} = 0$ if S and O are independent. Not conversely!
 - Indeed $\mathcal{L}_{SO} = 0$ if O never affects \mathcal{A} 's best guess.

25

Example ("base-rate fallacy")

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

	positive	negative
channel matrix		
cancer	0.90	0.10
no cancer	0.07	0.93

- A priori (age 40-50, no symptoms, no family history)
 $P_S[\text{cancer}] = 0.008$ $P_S[\text{no cancer}] = 0.992$

	positive	negative
joint matrix		
cancer	0.00720	0.00080
no cancer	0.06944	0.92256

column maximums

- $V(S|O) = 0.992 = V(S)$, so $\mathcal{L}_{SO} = 0$.
- Always guess "no cancer"! ($P[\text{cancer}|\text{positive}] \approx 0.094$)

26

Capacity

- **Min-capacity**, $\mathcal{ML}(C)$, is the maximum min-entropy leakage, over all a priori distributions P_S .
- Similarly, **Shannon capacity**.
- **Theorem:** $\mathcal{ML}(C)$ is the log of the sum of the column maximums of C .
- Also, $\mathcal{ML}(C)$ is realized by a uniform a priori P_S .
- **Corollary:** $\mathcal{ML}(C) = 0$ iff the rows of C are identical.
- **Corollary:** If C is deterministic, then $\mathcal{ML}(C)$ is the log of the number of feasible outputs.

27

Min-capacity and Shannon capacity

- **Theorem:** On deterministic channels, min-capacity and Shannon capacity coincide.
- On **probabilistic** channels, min-capacity can exceed Shannon capacity by an arbitrary factor:

2^{-10}	2^{-64}	2^{-64}	...	2^{-64}
2^{-64}	2^{-10}	2^{-64}	...	2^{-64}
2^{-64}	2^{-64}	2^{-10}	...	2^{-64}
...
2^{-64}	2^{-64}	2^{-64}	...	2^{-10}

Shannon capacity ≈ 0.05
 Min-capacity ≈ 54.00

- **Conjecture:** Shannon capacity cannot exceed min-capacity.

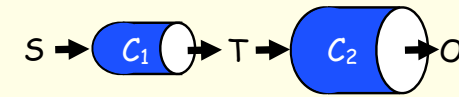
28

Plan of the talk

- Motivation
- Information-theoretic channels
- Quantifying leakage
 - using mutual information
 - using min-entropy
 - channel capacity
- Channels in cascade
 - application to timing attacks on cryptography
- Some techniques for calculating min-entropy leakage

29

Channels in Cascade [EspinozaSmith11]

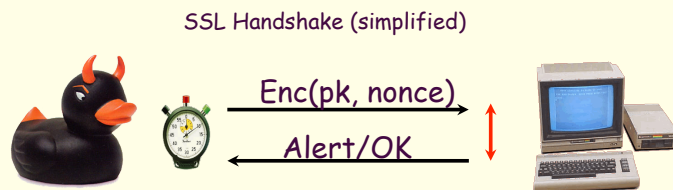


- Cascading of channels corresponds to multiplication of channel matrices $C = C_1 C_2$.
- **Theorem:** If C is the cascade of C_1 and C_2 , then for any P_S , $\mathcal{L}_{SO} \leq \mathcal{L}_{ST}$.
 - Analogue of the **data-processing inequality**.
 - Curiously, we **can** have $\mathcal{L}_{SO} > \mathcal{L}_{TO}$.
- **Theorem:** If C is the cascade of C_1 and C_2 , then we have $\mathcal{ML}(C) \leq \mathcal{ML}(C_1)$ and $\mathcal{ML}(C) \leq \mathcal{ML}(C_2)$

30

Application: timing attacks on cryptography

- Remote timing attack [Boneh Brumley 2003]
- 1024-bit RSA key recovered **in 2 hours** from standard OpenSSL implementation across LAN.



31

Effectiveness of blinding and bucketing against timing attacks [KöpfSmith10]

- **Blinding:** randomize ciphertext before decryption; de-randomize after decryption.
- **Bucketing:** force decryption to take one of a small number of possible times.
- **Theorem:** With blinding and bucketing, the number of min-entropy bits leaked is logarithmic in the number of timing observations.
 - Proved by factoring the channel matrix into a cascade where the set T of intermediate values is small.

32

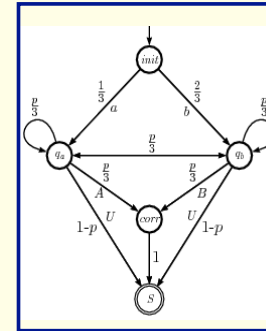
Plan of the talk

- Motivation
- Information-theoretic channels
- Quantifying leakage
 - using mutual information
 - using min-entropy
 - channel capacity
- Channels in cascade
 - application to timing attacks on cryptography
- **Some techniques for calculating min-entropy leakage**

33

Computing leakage by model checking techniques e.g. reachability analysis [Andrés et al. '10]

Crowds protocol as a probabilistic automaton



Linear equations

$$\begin{array}{l}
 x_{init}^{aA} = \frac{1}{3} \cdot x_a^A, \quad x_a^A = \frac{p}{3} \cdot x_a^A + \frac{p}{3} \cdot x_b^A + \frac{p}{3} \cdot x_{corr}^A, \quad x_{corr}^A = x_S^A, \\
 x_{init}^{bA} = \frac{2}{3} \cdot x_a^A, \quad x_b^A = \frac{p}{3} \cdot x_a^A + \frac{p}{3} \cdot x_b^A + \frac{p}{3} \cdot x_{corr}^A, \quad x_S^A = 0, \\
 x_{init}^{aB} = \frac{1}{3} \cdot x_a^B, \quad x_a^B = \frac{p}{3} \cdot x_a^B + \frac{p}{3} \cdot x_b^B + \frac{p}{3} \cdot x_{corr}^B, \quad x_{corr}^B = x_S^B, \\
 x_{init}^{bB} = \frac{2}{3} \cdot x_b^B, \quad x_b^B = \frac{p}{3} \cdot x_a^B + \frac{p}{3} \cdot x_b^B + \frac{p}{3} \cdot x_{corr}^B, \quad x_S^B = 0, \\
 x_{init}^{aU} = \frac{1}{3} \cdot x_a^U, \quad x_a^U = \frac{p}{3} \cdot x_a^U + \frac{p}{3} \cdot x_b^U + (1-p) \cdot x_S^U, \quad x_{corr}^U = x_S^U, \\
 x_{init}^{bU} = \frac{2}{3} \cdot x_b^U, \quad x_b^U = \frac{p}{3} \cdot x_a^U + \frac{p}{3} \cdot x_b^U + (1-p) \cdot x_S^U, \quad x_S^U = 1.
 \end{array}$$

Solution (assuming $p = 0.9$)
by Gaussian elimination

$$\begin{array}{l}
 x_{init}^{aA} = \frac{7}{40}, \quad x_{init}^{aB} = \frac{3}{40}, \quad x_{init}^{aU} = \frac{1}{12}, \\
 x_{init}^{bA} = \frac{3}{20}, \quad x_{init}^{bB} = \frac{7}{20}, \quad x_{init}^{bU} = \frac{1}{6}.
 \end{array}$$

		A	B	U
Joint matrix	a	7/40	3/40	1/12
	b	3/20	7/20	1/6

$$\begin{array}{l}
 V(S) = 2/3 \\
 V(S|O) = 7/40 + 7/20 + 1/6 = 83/120 \\
 \mathcal{L}_{S|O} = \log [(83/120)/(2/3)] = \log (83/80)
 \end{array}$$

34

Min-capacity of deterministic programs

- On deterministic programs, the min-capacity (and Shannon capacity) is the log of the number of feasible final values of O .
- Example (S and O are 32-bit unsigned integers):

```

S = S & 0x77777777;
if (S <= 64) O = S; else O = 0;
if (O % 2 == 0) O++;
    
```

- O has 17 feasible values:
1, 3, 5, 7, 17, 19, 21, 23, 33, 35, 37, 39, 49, 51, 53, 55, 65
- Hence the min-capacity is $\log 17 \approx 4.087$ bits.

35

Bounds using two-bit patterns [MengSmith11]

- Determine **patterns** that the bits of O must satisfy.
- Single bits can be either **Zero**, **One**, or **Non-fixed**.
- On the example, the one-bit patterns are
000000000000000000000000***0**1
- Pairs of Non-fixed bits satisfy one of seven relations: **Eq**, **Neq**, **Geq**, **Leq**, **Nand**, **Or**, **Free**.
- On the example, we get four non-Free patterns:
Nand(6,5), Nand(6,4), Nand(6,2), Nand(6,1)
- The number of satisfying assignments to the bit patterns is an upper bound on the number of feasible outputs. (Here it's 17, which is exact.)

36

Conclusion and Future Directions

- Min-entropy leakage is an attractive foundation for the quantitative analysis of confidentiality.
 - Can techniques for calculating leakage be scaled up to large systems? Are there compositional analyses?
 - What leakage policies should be enforced?
 - How do min-entropy leakage and **differential privacy** fit together?
- Thanks to my collaborators:
Catuscia Palamidessi, Miguel Andrés, Boris Köpf,
Ziyuan Meng, Barbara Espinoza

37

Questions?



38