Thermodynamic Aspects of Confidentiality

Pasquale Malacaria*, Fabrizio Smeraldi

School of Electronic Engineering and Computer Science, Queen Mary University of London, Mile End Road, London E1 4NS, UK

*Corresponding author.

Email addresses: pm@eecs.qmul.ac.uk (Pasquale Malacaria), fabri@eecs.qmul.ac.uk (Fabrizio Smeraldi)

Thermodynamic Aspects of Confidentiality

Pasquale Malacaria*, Fabrizio Smeraldi

School of Electronic Engineering and Computer Science, Queen Mary University of London, Mile End Road, London E1 4NS, UK

Abstract

We analyse secure computation as a physical process and connect it to recent advances in security, namely Quantitative Information Flow.

Using a classic thermodynamic argument involving the second principle and reversibility we show that any deterministic computation, where the final state of the system is observable, must dissipate at least $Wk_BT\ln 2$. Here W is the information theoretic notion of security as defined in Quantitative Information Flow, k_B the Boltzmann constant and T the temperature of the environment. Such minimum dissipation is also an upper bound on another probabilistic quantification of confidentiality introduced by Smith.

We then explore the thermodynamics of timing channels in Brownian computers. Here the low energies involved lead to the emergence of new timing channels arising directly from the entropy variations related to computation.

Keywords: Information Theory, Security, Thermodynamics

1. Introduction

The study of computation as a physical process connects to a long tradition straddling the fields of physics and computer science. Thermodynamic aspects of computation have been considered, among others, in the works of Bennett [4], Feynman [14], Landauer [21]. One of the main achievements is arguably the realisation that computation, per se, does not require any energy dissipation [4]. Key to this result is the demonstration that any computation can in principle be embedded into a reversible computation, and these can be carried out at no energy cost.

However there are application areas of great practical importance that are concerned with intrinsically irreversible computations, confidentiality being the foremost among them. In this work the intuitive connection between the physics of irreversible computation and confidentiality is made formal. Instrumental to

*Corresponding author.

Email addresses: pm@eecs.qmul.ac.uk (Pasquale Malacaria), fabri@eecs.qmul.ac.uk (Fabrizio Smeraldi)

establishing a formal connection is recent research in confidentiality under the name of Quantitative Information Flow [11, 12] where the leakage of confidential information is quantified in terms of information theory.

Quantitative Information Flow (QIF) was introduced to address over-restrictive definitions of confidentiality: ideally a secure system ought to be able not to disclose any confidential information. In practice no usable system has this desirable "zero leakage", or non-interference property. Any password protected system leaks some information to an attacker even by refusing access to the system (the attacker will then learn that the password is not the one attempted).

Because of the unavoidability of leakage QIF provides an alternative approach to confidentiality: it aims to measure the leakage and so to provide support for a risk assessment of the security threat. Measuring leakage is achieved by measuring the information about the secret data an attacker can infer by observing the system. For example, attempting to randomly guess a pin number at an ATM machine will generate two possible observations: (1) the pin is accepted (probability of acceptance 0.0001), (2) the pin is rejected (probability of rejection 0.9999). A standard measure of information is Shannon's entropy. When evaluated on these probabilities it allows the inference that the attacker has gained 0.00147 out of the total 13.2 bits of information about the secret pin in this attack: an insignificant leak unless the attack is repeated multiple times. More generally, given an initial distribution on the confidential data and a deterministic program P whose sole input is the confidential data, the leakage is defined as the Shannon entropy of the probability distribution associated to possible observable outputs. This definition is consistent with the naive "zero leakage" definition: it is easy to prove that a program leaks no confidential information if and only if its output has zero entropy [11].

Quantitative Information Flow has been applied among others to side channel attacks analysis [17, 18, 9], to measure confidentiality leaks in the Linux Kernel [20], to database security analysis [2], to the analysis of anonymity protocols [6, 7], to side channels leaks in web applications [32] and the avoidance of fault masking [10].

1.1. Contributions

In the first part of the paper (Sections 4 and 5) we consider quasi-static transformations. This setting allows us to prove fundamental bounds valid for a generic computational system. These bounds were first published in [23].

In Section 4 we start from the second principle of thermodynamics and its application to the erasure of information (Landauer principle). Using a classical thermodynamic argument based on reversibility, we prove a lower bound on dissipation for secure computation. Up to the multiplicative factor $k_B T \ln 2$, this bound is given by the notion of security W from Quantitative Information Flow (equation 18 and proposition 3).

Section 4.4 demonstrates that net energy expenditure is not required if we allow probabilistic operators into the language. In that case work can actually be extracted by the system (inequality 21), although erasure remains an irreversible

operation. Again the energy is bounded by the quantity W which is in this case non-positive.

Section 5 investigates the thermodynamics of Smith's definition of leakage and proves that remaining uncertainty is in general a lower bound on W (proposition 6). The bound becomes an equality if and only if the remaining uncertainty coincides with the difference between the work needed to reset the input and output registers when they are in their maximally disordered state (proposition 4 and 7). To the best of our knowledge this is the first connection between guessability and thermodynamics. Finally both measures are order related to the magnitude of dissipation in section 5.1.

In order to study timing channels the second part of the paper (Sections 6, 7) deals with a class of computers with low energy requirements and governed by a specific dynamics, namely Brownian computers. Following classic arguments from Feynman and Bennett we investigate energy dissipation in such computers and the relation between energy and speed of computation (section 6.3). We then apply these arguments to timing channels (section 7) and show the existence of length-of-computation timing channels (section 7.1) as well as of a novel class of entropy timing channels (section 7.2). Trade-offs between these channels are illustrated in section 7.3.

2. Background

2.1. Notations

Given probabilities $\mu_1, \ldots, \mu_N$, the Shannon entropy of the distribution is defined as

$$H(\mu_1, \ldots, \mu_N) = -\sum_{1 \leq i \leq N} \mu_i \log(\mu_i) \tag{1}$$

where log denotes the logarithm in base 2. The entropy of a random variable X is

$$H(X) = -\sum_{X} \mu(X = x) \log(\mu(X = x)).$$

Given two random variables X, Y the conditional entropy $H(X|Y)$ is defined as $H(X,Y) - H(Y)$ where $H(X,Y)$ is the entropy of the joint random variable (X,Y). $H(X|Y)$ measures the uncertainty on X knowing Y.

Mutual information is defined as $I(X;Y) = H(X) - H(X|Y)$ and conditional mutual information as $I(X;Y|Z) = H(X|Z) - H(X|Y,Z)$. Mutual information is a measure of the correlation between X and Y, i.e. it quantifies how much information they share.

2.2. Basic definitions and properties

A general definition of *leakage* assumes an information processing system having inputs h, l where h are the *confidential* inputs and l are the public inputs and a set of *observables* P probabilistically related to the inputs. The leakage of confidential data h to the observables P given public input l is then defined as the

difference in the uncertainty about the secret before and after the observations and is measured using conditional mutual information [11]:

$$I(h; P|l) = H(h|l) - H(h|P, l) \tag{2}$$

In simple terms this is what the attacker has learned about the secret by observing the system.

In the case of a deterministic program where the sole input is h and observations P are the outputs, definition (2) reduces to mutual information $I(h; P)$ and the following holds:

$$\begin{aligned} I(h; P) &= H(h) - H(h|P) && (3)\\ &= H(P) - H(P|h) && (4)\\ &= H(P) && (5) \end{aligned}$$

where the second equality holds because mutual information is symmetric and the third equality holds because the outputs of a program only depend on h, hence $H(P|h) = 0$.

Notice that as discussed in [24] (resp. [20]) the restriction to h being the sole input is not a limitation in the theory (resp. in practice). In this work we will assume that the final memory state of the system is observable.

The key quantity in this paper is $W = H(h) - H(P)$.

Proposition 1. *For deterministic programs with sole input h the following are equivalent:*

1. $W = H(h) - H(P)$
2. $W = H(h) - I(h; P)$
3. $W = H(h|P)$

The equivalences follow easily from equations 3, 4 and 5.

In its formulation $W = H(h|P)$ the quantity W is known in the Quantitative Information Flow community as *security* [24, 11]. The reason why the first definition of W is the chosen one is related to its generality beyond deterministic systems and will be clarified in section 4.4.

Consider now the equivalent formulation

$$W = H(h) - I(h; P).$$

In words this says that W is the amount of secret that has not been leaked by the program, i.e. the secret *protected* by the program.

A contribution of this paper is to show that $W \ln(2) k_B T$ represents also the thermodynamic work to be done *on* the system to protect the confidential data (equation 18). In other words $W \ln(2) k_B T$ is the minimum amount of energy any system implementing that program must dissipate in order to protect that confidential data.

To understand the basic properties of W it helps to introduce equivalence relations induced by the observations: we say that two confidential values are equivalent if the program will produce the same output when given those inputs [24, 20].

Proposition 2. *The maximum and minimum of W are as follows:*

1. *W is maximal for the distribution on the secret which is uniform on the largest equivalence class and 0 on all other points. This maximal value is $\log(|e|)$ where e is the largest equivalence class of confidential values and $|e|$ its cardinality.*
2. *W is minimal for the distribution on the secret which is 0 everywhere apart for only one point in each equivalence class (and is uniform on these points). The minimal value is 0.*

As a sanity check consider $W = H(h|P) = 0$: that means that an attacker will have no uncertainty about the secret given the observations, hence everything has been leaked i.e. no work has been done to protect the secret. This case also cover the situation where the program has no confidential input: these are computations with no security constraints and so reversible computations in the sense of Bennett [4].

At the other end W is maximal when $H(h|P) = H(h)$, that is when the observations and the secret are independent - hence all bits of the secret have been protected. A particular case of this is the computation of a constant function.

3. The thermodynamics of computation

While a thorough review of the physics of computation is beyond the scope of this work, we believe that an overview of the salient points of this discussion can actually help the reader locate our contribution in the wider subject area.

3.1. Modelling computation

Most of the key conclusions in the thermodynamics of computation can be arrived at through the analysis of relatively simple physical models of computation involving idealised objects such as perfectly rigid spheres, a single molecule of a perfect gas, or quantum systems with few degrees of freedom (in the case of quantum computers). A useful physical model of computation is given by a set of (idealised) billiard balls arranged in a particular way, that are set into motion starting from an input condition and will eventually evolve through a series of perfectly elastic collisions to a configuration representing the output state. For the sake of this argument, we can assume that the presence of a ball in a particular position at the beginning (or respectively the end) of a computation represents a 1 state, while its absence represents a 0 state. A suitably complex system of billiard balls can in principle carry on any computation [16, 14], with some qualifications that will become clear below.

3.2. Time-symmetry and reversibility

The most important feature of the billiard-ball model is its reversibility, that directly derives from the symmetry of the laws of mechanics with respect to time. Concretely, the total energy of the balls is conserved during a computation; it is then sufficient to reflect the balls backwards into the computer at the end of computation for this to be undone as the balls return to their starting position. Since the position of the balls encodes the state of the computer, this implies that all the functions computed are logically reversible — that is, one-to-one — and they are computed at zero energy cost. More generally, the time symmetry of physical laws and the existence of universal reversible gates such as the controlled-XOR gate introduced by Friedkin and Toffoli [16] imply that all logically reversible functions can be computed reversibly without energy expenditure. Since a non-injective function $y = f(x)$ can easily be made invertible by enriching the output with the input ($\tilde{f}(x) = (x, f(x))$), all computations can in principle be done reversibly and without any minimum energy expenditure; however, there are evidently cases (security being one of them) where it is clearly not desirable to do so.

3.3. The Second Principle and irreversibility

Irreversibility arises in Thermodynamics from the statistical study of a high number of copies of the same system. This gives rise to one of the most powerful and all-encompassing concepts of a time arrow, encoded in the Second Principle. The Second Principle associates to each system a state function $\mathcal{S}$ known as entropy, and states that when the system undergoes a transformation, the following inequality holds:

$$\Delta\mathcal{S} \geq \frac{\delta Q}{T} \tag{6}$$

where δQ is the heat *absorbed* by the system at temperature T and the equality sign holds for reversible transformations only. Since entropy is a function of the state of the system, this means that an isolated system (that cannot dump heat into the environment) will tend to evolve irreversibly towards states with higher entropy. For a computer, that is generally modelled as being in equilibrium with a single heat source at temperature T (the environment), Equation 6 implies that if the machine is returned to its initial state ($\Delta\mathcal{S} = 0$) after an *irreversible* transformation, a certain amount of energy is dissipated as heat into the environment during the process:

$$0 = \Delta\mathcal{S} > \oint \frac{\delta Q}{T} = \frac{1}{T} \oint \delta Q = \frac{\Delta Q}{T} \tag{7}$$

as the inequality sign will strictly hold in (6) during at least part of the transformation. Since the computer is reverted to its initial state, the energy dispersed as heat must be compensated by doing an equal amount of work on the system.

In terms of the microstates of the system, i.e. of a complete specification of all its degrees of freedom, entropy can be written as

$$\mathcal{S} = -k_B \sum_i p_i \log p_i, \tag{8}$$

which bears a striking analogy to the Shannon entropy H in Equation 1. Indeed, since logical states in a computation are in one-to-one correspondence with the physical states of the computer, Equation 6 provides a direct way to relate changes in the information content of a register of the computer to energy consumption. In particular, any reduction of the information content of the register (for instance, a reset operation) will result in a negative $\Delta\mathcal{S}$ and thus require heat to be dispersed into the environment - and work to be done on the system if conservation of energy is to hold. The quantitative relation between the erasure of information and dissipation is beautifully brought out by a computational take on a puzzling conceptual experiment, i.e. Maxwell's demon. We will briefly review this argument in the next section.

3.4. From Maxwell's demon to the Landauer principle

Another consequence of Equation 6 is that, since $\Delta\mathcal{S} = 0$ over a transformation that ultimately reverts the system to its initial state, it is impossible to build a thermal machine that has as its only effect the transformation of energy from a single source of heat into work — in order to balance the entropy cheque, some heat will need to be dumped into a reservoir at lower temperature. This is known as the Kelvin statement of the Second Principle, and its hypothetical violation as a Perpetual Motion of the second kind.

An intriguing conceptual attempt on the Kelvin statement was produced by Maxwell with his demon. Maxwell considered a simple system consisting of a perfect gas contained in two chambers communicating via a trap door in the partition. The trap door is operated by a hypothetical agent (the demon) that, by cleverly opening and closing it, is able to group the fastest molecules into one side of the partition, thus creating a pressure difference that can then be used to produce work for free. As the process can be repeated at will, this would be a perpetual motion of the second kind. Various attempts have been made at exorcising the demon, notably focusing on the cost to the demon of measuring the position and speed of the particle prior to making a decision on opening the trap door, or on the temperature and thermal agitation of the demon itself. However, the modern consensus is that measurements can be performed at arbitrarily low cost [22]. Rather, the demon itself is viewed as a computing machine that must have at least one bit of memory — in order for it to know whether it should open the trap-door to let the particle through or not. Safeguarding the Second Principle requires that the cost for the demon of resetting its memory to prepare it for another run is precisely $k_B T \ln 2$.

This compelling argument about the interaction between information and physical entropy is given by Bennett and illustrated by the simplified version of Maxwell's argument shown in figure 1. The system consists of a single particle in a box. In the initial state (1) the observer (demon) is completely ignorant on the whereabout of the particle (see right side of figure). Next (2) the observer, at cost zero, inserts a partition separating the box in two chambers. He now knows for sure that the particle is on the left hand side or on the right hand side but he doesn't know which one. In the next step (3) he observes the box and so acquires information on the position of the particle (left or right side); this

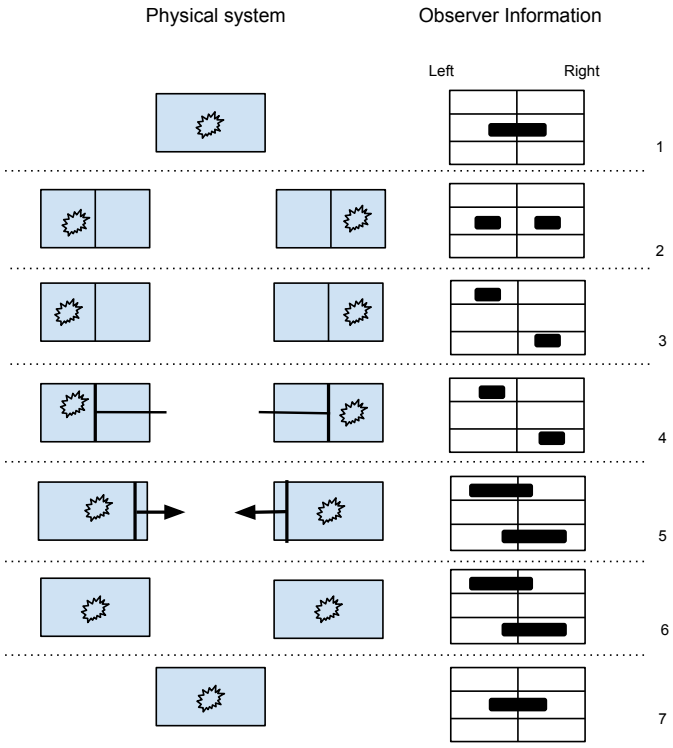


Figure 1: Interaction between information and physical entropy

observation requires no energy expenditure. Using this knowledge and again at cost zero (4) he can then push a piston in the empty chamber up to the partition. Next (5), again at cost zero, he can remove the partition and let the pressure exerted by the particle push the piston: the work generated by the system by pushing the piston is $k_B T \ln 2$. His knowledge is now that the particle was on the left hand side (or the right hand side). Eventually the expansion process terminates (6) and the box is seemingly back to the initial state. So far $k_B T \ln 2$ energy has been extracted from the system. The only difference between the initial state of the system (1) and this sixth step of the process is that the observer still has information about the origin of the particle (1 bit of information). Hence resetting the observer's memory to the initial state of complete ignorance (7) $\equiv$ (1) must require minimum work $k_B T \ln 2$ to compensate for the work extracted in step (5).

This is an instance of the celebrated **Landauer principle** [21]:

> The minimum cost for the cancellation of one bit of information is $k_B T \ln 2$.

This principle has very recently also been experimentally demonstrated [31]. As argued by Bennett [4], Feynman [14], this intrinsic cost of cancelling information is the key consideration in the thermodynamics of computation.

4. Physical model of secure computation

4.1. A simple two state register

In this section, we derive a few basic results on the energetic cost of erasing information using a simple and rather idealised physical model. While having a specific model is useful to understand the type of reasoning involved, our final results do not depend on the model and have quite general applicability.

In its simplest version, our model of a one-bit system consists of one molecule of a perfect gas contained inside a box divided in two chambers by a partition. The two chambers are labelled with the states 0 and 1; the system is in thermal equilibrium with a heat reservoir at temperature T. If the particle has equal probability of being in either chamber, we can reset the system by removing the central partition and use an (idealised) piston to compress the gas into the chamber marked 0. For a perfect gas, $\mathcal{P}V = nk_BT$, where n is the number of molecules, V the volume and $\mathcal{P}$ the pressure. We assume that the two chambers have unit volume. In this case, the work done *by* the system during compression is

$$\int_2^1 \mathcal{P}dV = k_BT \int_2^1 \frac{1}{V}dV = -(\ln 2)k_BT \tag{9}$$

that agrees with Landauer's principle.

A more interesting case is obtained if we assume that the particle is found in the two chambers with different probabilities μ_1 and μ_2 (we assume without loss of generality that $\mu_1 > \mu_2$). It is useful to consider an ensemble of identical boxes, each containing a single molecule of an ideal gas. The molecule is in the left half of the box in proportion μ_1 of the boxes and in the right half in proportion μ_2. Assume that the partition of each box is actually a piston initially placed in the centre position, and that all the shafts are joined together. Since more particles will hit one of the pistons on the left hand side than on the right hand side, the pistons will move to the right until the pressure on both sides is equalised. This expansion can be used to extract work from the system, leaving it in the maximally disordered state; the energy thus obtained can then be offset against the work needed for a reset. Figure 2 illustrates the idea for a system consisting of a two states with probabilities 1/3, 2/3.

Again assuming that each chamber initially has unit volume, and averaging across the ensemble, we have $\mathcal{P}_i = \mu_i k_BT$, with $\mathcal{P}_1$ being the pressure in the left chamber and $\mathcal{P}_2$ the pressure in the right chamber. The volumes of the chambers at the end of the expansion obey $\mu_1/V_1 = \mu_2/V_2$, which is the condition for the pressure to equilibrate. Since $V_1 + V_2 = 2$, we have $V_1 = 2\mu_1$ at the end of the expansion.

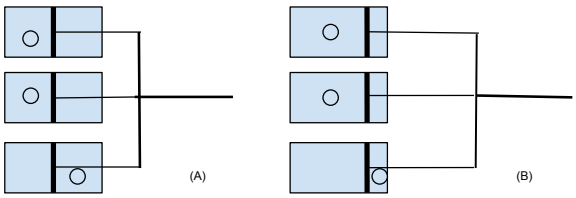


Figure 2: A two–state system with probabilities 1/3 and 2/3 before (A) and after (B) the expansion.

Therefore, the work done by the system during expansion is:

$$\begin{aligned}\frac{W_{exp}}{k_BT} = \frac{1}{k_BT}\int_1^{2\mu_1}(\mathcal{P}_1 - \mathcal{P}_2)dV = \\ = \int_1^{2\mu_1}\frac{\mu_1}{V}dV - \int_1^{2\mu_1}\frac{\mu_2}{2-V}dV = \\ = \mu_1 \ln 2\mu_1 - \mu_2 \ln\frac{2-1}{2-2\mu_1} = \\ = \mu_1 \ln 2\mu_1 + \mu_2 \ln 2\mu_2 = \\ = \mu_1 \ln 2 + \mu_1 \ln \mu_1 + \mu_2 \ln 2 + \mu_2 \ln \mu_2 = \\ = \ln 2 - H(\mu_1, \mu_2)\ln 2 \quad (10)\end{aligned}$$

where for convenience we have divided both sides by k_BT.

After the expansion we can reposition at no cost the pistons at one end of the combined chambers and we are left to reset the same maximally disordered system we considered above, which according to Equation 9 can be done at a cost $(\ln 2)k_BT$. We conclude that the work needed to reset a two–state system with probabilities μ_1, μ_2 is

$$(\ln 2)k_BT - W_{exp} = H(\mu_1, \mu_2)k_BT \ln 2.$$

4.2. The multiple–state case

We now introduce a generalisation of the above perfect gas model to an $N-$state system, able to represent N distinct logical states with probabilities $\mu_1 \ldots, \mu_N$. We will use this conceptual model to compute the work required to reset the representation of an arbitrary distribution of logical states.

Our generalised physical model consists of a box with N chambers, each initially of unit volume. The partitions of the chambers are pistons attached to separate shafts that can be actuated independently. Figure 3 illustrates the idea. The box contains exactly one molecule of ideal gas, that is found in the i-th chamber with probability μ_i (again, it is useful to think of an ensemble of such boxes in a fraction μ_i of which the particle is found in chamber i).

We again assume, for convenience, that the chambers are arranged in order of decreasing probability of containing the particle (the general case can be treated in a similar way by letting the pistons expand in different predetermined directions). In order to reset the system we start by performing a series of reversible expansions between adjacent cells, followed by removing the partitions between cells that have been brought into equilibrium. Specifically, we begin by expanding the first (leftmost) chamber against the second (storing the work done in the process somewhere). We then remove the partition between the first two chambers and expand the resulting joint volume against the third chamber. This process is iterated until the system is brought to its maximally disordered state and equilibrium is reached; energy produced by all expansions can then be used to help resetting the system to its initial state.

We shall now work out a generic stage in this expansion, namely the expansion of the cells numbered 1 through $n-1$ (that we suppose have already been merged) against cell n.

Let the cumulative probability of the particle being in cell 1 through n be $M_n = \sum_{i=1}^{n} \mu_i$. Hence the pressure in the first $n-1$ chambers after the partitions between them have been removed is

$$\mathcal{P}_{n-1} = M_{n-1} k_B T/(n-1),$$

$n-1$ being the volume.

Let $\mathcal{P}_n$ be the pressure of the next individual chamber, i.e.

$$\mathcal{P}_n = \mu_n k_B T.$$

After the expansion, the volume of the n-th cell will be a fraction μ_n/M_n of the total volume of the n cells, which we assume is n.

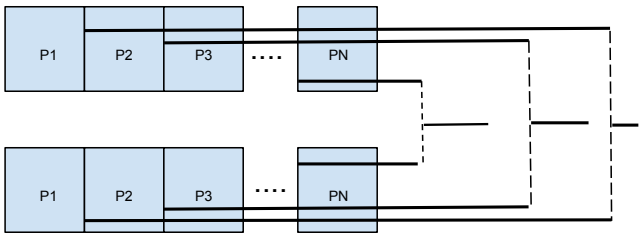


Figure 3: Modelling a system with N states. The average pressure $\mathcal{P}_i$ in each chamber is proportional to the probability μ_i of finding the particle in that chamber.

Thus work done during the expansion, similarly to equation 10 is

$$\frac{W_n}{k_BT} = \frac{1}{k_BT}\int_{n-1}^{n(1-\mu_n/M_n)} (\mathcal{P}_{n-1} - \mathcal{P}_n)dV =$$
$$= \int_{n-1}^{n(1-\mu_n/M_n)} \left(\frac{M_{n-1}}{V} - \frac{\mu_n}{n-V}\right) dV =$$
$$= M_{n-1}\ln\frac{n(1-\mu_n/M_n)}{n-1} + \mu_n \ln\frac{n\mu_n}{M_n} =$$
$$= M_{n-1}\ln\left(\frac{n}{n-1}\frac{M_{n-1}}{M_n}\right) + \mu_n\ln\frac{n\mu_n}{M_n} =$$
$$= M_n\ln\frac{n}{M_n} - M_{n-1}\ln\frac{n-1}{M_{n-1}} + \mu_n\ln\mu_n \quad (11)$$

The work extracted from the system during the series of expansions can now be obtained as the sum of the contribution of all the pairwise expansions:

$$\frac{W_{exp}}{k_BT} = \sum_{n=1}^{N}\frac{W_n}{k_BT} =$$
$$= \sum_{n=1}^{N}\left(M_n\ln\frac{n}{M_n} - M_{n-1}\ln\frac{n-1}{M_{n-1}}\right) + \sum_{n=1}^{N}\mu_n\ln\mu_n. \quad (12)$$

Noticing that the term in brackets yields a telescopic sum and that $M_N = \sum_{i=1}^{N}\mu_i = 1$ we obtain

$$\frac{W_{exp}}{k_BT} = M_N\ln\frac{N}{M_N} + \sum_{n=1}^{N}\mu_n\ln\mu_n = \ln N + \sum_{n=1}^{N}\mu_n\ln\mu_n. \quad (13)$$

This represents all the work extracted from the system during the expansion, that leaves it in the maximally disordered state — i.e. with the particle equally likely to be in any of the chambers. At this point, resetting the system to the initial state requires the following work:

$$\frac{W_{comp}}{k_BT} = \int_N^1 \frac{1}{V}dV = \ln N \quad (14)$$

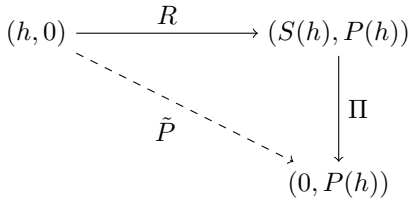
Thus the net work done on the system to reset it form an arbitrary distribution of states $\mu_1, \mu_2, \ldots, \mu_N$ is

$$W_{reset} = W_{comp} - W_{exp} = \quad (15)$$
$$= -\sum_{n=1}^{N}\mu_n\ln\mu_n k_BT = \quad (16)$$
$$= H(\mu_1\ldots,\mu_N)k_BT(\ln 2) \quad (17)$$

4.3. Universal factorisation of secure computations

Bennett [4], Friedkin and Toffoli [16] demonstrated that computations can in principle be performed reversibly, hence there is no need for dissipation in the computational process. Consistently with Bennett's ideas we factor a secure computation $\tilde{P}$ into a reversible computation R and a resetting step Π.

We build the following commutative diagram:



i.e. $\tilde{P} = \Pi \circ R$ (we will, in the following, identify $\tilde{P}$ with P where no confusion can arise). In the above diagram the extra register $S(h)$ holds the history, i.e. the information required for reversing the calculation R. After R terminates, Π enforces security by deleting the history $S(h)$.

Figure 4 illustrates this process for the program `l=h%2;` with `h` being a two–bit secret.

Note that there is a wide choice in the implementation of S, and thus of the two programs R and Π. An obvious choice is for S to hold a copy of the input (which is generally not minimalistic, as our example in Figure 4 shows). However, as we will see the particular implementation of S does not affect the energy cost of the computation. Indeed, since S is needed to disambiguate between input states h_i, h_j leading to the same program output P_k the only requirement on S is that for each equivalence class in the observational equivalence S is one-to-one. Thus given a program outcome $P_k = P(h_i)$, the probability of the associated history $S(h_i)$ is equal to the probability of the input h_i, i.e. :

$$\mu(S(h_i)|P_k) = \mu(h_i|P_k).$$

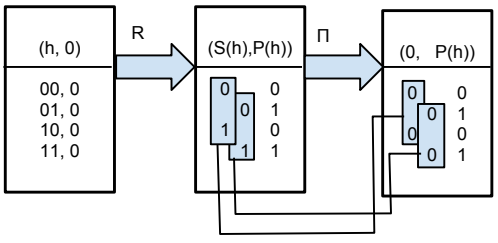


Figure 4: Secure computation of `l = h%2` on a two-bit secret. The shaded history register $S(h)$ is reset by Π in the last step.

Combining this with equation 15 it then follows that the cost of resetting S, averaged over all program outputs, is

$$\begin{aligned}\left(\sum_k \mu(P_k) \sum_{h\in P^{-1}(P_k)} \mu(h|P_k) \ln \frac{1}{\mu(h|P_k)}\right) k_B T &= \\ = H(h|P) k_B T \ln 2 &= \\ = W k_B T \ln 2 & \qquad (18)\end{aligned}$$

that is the energy equivalent of the *security* of the program.

This result is universal i.e.

Proposition 3. *$W k_B T \ln 2$ is a lower bound on the energy dissipated by any system implementing $\tilde{P}$.*

To prove it suppose an implementation $\tilde{P}_0$ dissipates less than $W k_B T \ln 2$, then $\tilde{P}_0 \circ R^{-1}$ is effectively an implementation of the reset operation Π that violates Landauer's principle.

4.4. Erasure vs resetting: extracting work from the system

Security imposes a lower bound on dissipation only in the case of deterministic computation, in which the system is reset to a fixed state. An alternative process for protecting confidential data consists in overwriting the information to be kept confidential with randomly generated bits; we call such cancellation erasure by randomisation (in this section simply erasure). Considering the graph in Section 4.3, we replace the reset operator Π with an erasure operator E:

$$(S(h), P(h)) \stackrel{E}{\longmapsto} (\epsilon, P(h)) \qquad (19)$$

where ϵ is a random number. Alternatively, the erasing program is given by $\tilde{P}_e = E \circ R$, where R performs the computation reversibly and E assigns random bits to the register $S(h)$ containing confidential data. Notice that the deterministic model of computation has now been extended with a probabilistic operation E (we comment on this below).

We now have

$$H(\tilde{P}_e) = H(P) + \log(|S(h)|) \qquad (20)$$

where the second term is the entropy of generating a random string of size $|S(h)|$ (i.e. $|S|$ is the length in bits of register S). Therefore

$$\begin{aligned} W &= H(h) - H(\tilde{P}_e) = \\ &= H(h) - H(P) - \log(|S(h)|) \leq 0 \qquad (21)\end{aligned}$$

The second equality is illustrated by the commutative diagram in Figure 5: we know from section 4.3 that Π has cost $H(h) - H(P)$ and Π' has, by Landauer principle, cost $\log(|S(h)|)$. Inequality 21 then follows because $S(h)$ and $P(h)$

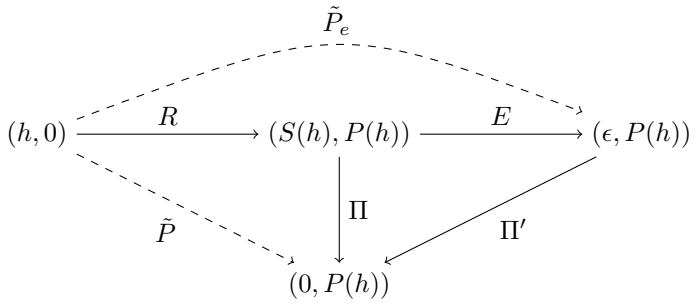


Figure 5: Relation between erasure and resetting

together have the same information content as h, and $\log(|S(h)|)$ is an upper bound on the information content of $S(h)$.

If the inequality is strict then W is negative, meaning that work can be *extracted* from the system; such work results from the randomisation and consequent increase in entropy of the history register S (note that the length of S can be arbitrary). However, W will be zero if the computation R already leaves S in a maximally disordered state — in which case further randomisation does not allow us to extract any work from the register. It should also be noted that, according to the Landauer principle, work extracted from the system during erasure will have to be paid back should one decide to revert the system to its original state (for instance to allow further use).

An important remark about erasure is that the introduction of probabilistic operators like erasure means that the leakage is no longer correctly described by the entropy of the observables $H(\tilde{P}_e)$. In fact, the term $\log(|S(h)|)$ in equation 20 should not count towards leakage as it corresponds to disorder injected into the system by the erasure operator. For this reason the general definition of leakage is given in terms of mutual information (equation 2); in fact

$$\begin{aligned} I(P_e;h) = H(P_e) - H(P_e|h) &= \\ = (H(P) + \log(|S(h)|)) - \log(|S(h)|) &= H(P) \end{aligned}$$

Here $H(P_e|h) = \log(|S(h)|)$ because the output of the program P is known when h is given; hence the only uncertainty comes from the randomisation of S.

Probabilistic operators are also the reason why we defined W as $H(h)-H(P)$ instead of setting $W = H(h|P)$. While the two definitions are equivalent in the deterministic setting they differ in the probabilistic one. In fact by choosing $W = H(h|P)$, as the conditional entropy is always non-negative we would conclude that dissipation is needed to protect confidential data also in the case of probabilistic systems; however we have just shown that it is possible to extract work from systems in non-maximally disordered states (an alternative argument for this uses Bennett's fuel value of information [14, 4]). Since this work can exceed the work needed to protect confidential data, $H(h|P)$ would be an

imprecise definition.

5. The thermodynamics of min-entropy leakage

A known issue with Shannon's entropy as a measure of program security is its mismatch with guessability: random variables may have arbitrarily high entropy and still be highly likely to be guessed. This issue has prompted researchers in security to investigate alternative foundations for Quantitative Information Flow, with Geoffrey Smith [28] providing a notion based on the probability of guessing the secret in one try that we will now explore. As we here show, Smith's definition is also closely related to the energetic cost of deleting information and hence to the Landauer principle, although the focus is now on the input and output registers rather than on the information required to reverse the computation.

Smith quantifies the loss of confidentiality in terms of the difference between the (log of the) probability of guessing the secret before and after observing the output of a program. The logic underlying this approach is illustrated by the following two programs:

A `if` (`h`%8 == 0) `then` `x` = `h`; `else` `x` = 1;

B `x` = `h`& $0^{7k-1}1^{k+1}$;

Program A returns the value of h when the last three bits of the secret are 0, and returns 1 otherwise. Program B copies the last $k+1$ bits of the secret to the public variable `x` (& is the bitwise `and`).

Given a uniformly distributed secret h of size $8k$ bits (where k is a parameter), the two programs have very similar leakage ($H(A) = k + 0.169$, $H(B) = k + 1$); as we have seen, a very similar amount of work is thus needed to protect the secret (in both cases $W \simeq (7k-1)k_BT$). However, the two programs have an entirely different guessing behaviour. Program A discloses the whole secret with probability 1/8 (and very little otherwise), while program B always reveals the last $k+1$ bits of the secret — but we are then left to guess the remaining $7k-1$ bits with probability $1/2^{7k-1}$. As k is increased it gets a lot easier to guess the secret in one try after running program A than after running program B; conversely, the difference in the energy dissipated by each program becomes negligible.

For these reasons, Smith suggests a measure of confidentiality based on Renyi min-entropy [26]. The leakage of a program is defined as the difference between the a priori Renyi min-entropy:

$$H_\infty(h) = -\log(\max_{h_i \in h} \mu(h_i))$$

and the a posteriori Renyi min-entropy $H_\infty(h|P)$, expressed as a min-entropy conditioned over all possible values of the observables:

$$H_\infty(h|P) = -\log\left(\sum_{P_j \in P} \mu(P_j) \max_{h_i \in h} (\mu(h_i|P_j))\right).$$

As shown in [19] $H_\infty(h|P)$ is the log of the complement of the Bayes risk and is also called *remaining uncertainty*. In the case of our examples, $H_\infty(h|P)$ is $\simeq 8k-3$ for A and $k+1$ for B: a fitting quantification of the difference in guessability between the two programs.

It makes sense to try and understand the thermodynamic meaning of $H_\infty(h|P)$. If the security $W = H(h|P)$ is the minimum dissipation what, if anything, is $H_\infty(h|P)$ in thermodynamic terms?

A first connection is given by the following result:

Proposition 4. *For a deterministic program with a uniformly distributed secret as its input,*

$$H_\infty(h|P) = \log(|h|) - \log(|P|).$$

The proof of the above is a consequence of the following facts [28]:

1. The channel capacity of the two measures coincides, i.e.

$$\max_{\mu(h)}(H_\infty(h) - H_\infty(h|P)) = \max_{\mu(h)}(H(h) - H(h|P))$$

2. $\max_{\mu(h)}(H_\infty(h) - H_\infty(h|P))$ is given by the uniform distribution on the input h
3. $\max_{\mu(h)}(H(h) - H(h|P))$ is given by the uniform distribution on the outputs of the program (this is also the channel capacity for Shannon leakage)
4. in both cases the maximum is equal to the log of the number of outputs of the program (denoted by $\log(|P|)$).

The thermodynamic interpretation of $H_\infty(h|P)$ hence is the difference between the maximal work needed to reset the initial state of the system (the input register) and the maximal work needed to reset the final state (the output register).

The following result easily follows from proposition 4:

Proposition 5. *For a deterministic program with a uniformly distributed secret as its input the following are equivalent:*

1. $H_\infty(h|P) = W$
2. *the outputs of the program are uniformly distributed*
3. *the observational equivalence relation consists of equivalence classes all of equal size*

These conditions are for example true of program B above, but not of program A. A class of programs satisfying these conditions are for example the ones computing $h\%n$ where n is a divisor of $2^{|h|}$.

If we relax the condition about the input being uniformly distributed then Smith's remaining uncertainty always underestimates dissipation i.e.

Proposition 6. *For all deterministic programs and any distribution on h:*

$$H_\infty(h|P) \leq W.$$

The result is already proven in [29] using the Sathi-Vardy bound. We provide an alternative argument which will be used in the proof of proposition 7. We prove that

$$W - H_\infty(h|P) = H(h) - H(P) - H_\infty(h|P) \geq 0$$

Let b_i denote the marginal probability of an equivalence class with $b_i = \sum_j h_{ij}$. Also, $h_i^\star = \max_j h_{ij}$.

The above inequality can then be written as

$$\sum_i b_i \log b_i - \sum_{ij} h_{ij} \log h_{ij} + \log \sum_i h_i^\star \geq 0.$$

An upper bound for the second term is given by

$$\sum_{ij} h_{ij} \log h_{ij} \leq \sum_{ij} h_{ij} \log h_i^\star = \sum_i b_i \log h_i^\star \tag{22}$$

so that it will suffice to prove

$$\sum_i b_i \log b_i - \sum_i b_i \log h_i^\star + \log \sum_i h_i^\star = \\ = \sum_i b_i \log \frac{b_i}{h_i^\star} + \log \sum_i h_i^\star \geq 0 \tag{23}$$

From Theorem 2.7.1 in Cover and Thomas [13] we have that

$$\sum_i b_i \log \frac{b_i}{h_i^\star} \geq \sum_i b_i \log \frac{\sum_i b_i}{\sum_i h_i^\star} \tag{24}$$

Replacing the first term in the second line of (23) with the above we have

$$\sum_i b_i \log \frac{\sum_i b_i}{\sum_i h_i^\star} + \log \sum_i h_i^\star \quad = \log \tfrac{1}{\sum_i h_i^\star} + \log \textstyle\sum_i h_i^\star \quad = 0$$

(since $\sum_i b_i = 1$). This concludes the proof.

We can now strengthen proposition 5 to precisely characterise when dissipation and $H_\infty(h|P)$ coincide:

Proposition 7. *$H_\infty(h|P) = W$ iff the input is uniformly distributed and the output is uniformly distributed.*

The proof follows from the following observations: if we relax the requirement of uniform distribution on the input then inequality 22 is strict. If we relax the requirement of uniform distribution on the output then inequality 24 is strict. Any of these will make proposition 6 a strict inequality.

5.1. Dissipation and Intrinsic Source Code Threat

A further interesting connection between both measures of confidentiality and thermodynamics is given by considering the following problem: judging only from the source code, which of two programs P, P' is more of a confidentiality threat? One way to look at this problems is to argue that if we only know the source code we shouldn't make assumptions on any particular a priori distribution on h, so it is natural to define the ordering $P \leq_H P'$ iff for all possible a priori distributions on h, P leaks less than P'. In terms of Shannon's entropy we formalise this by

$$P \leq_H P' \iff \forall \mu_h.\ H(P) \leq H(P').$$

where μ_h ranges over all distributions on h.

Similarly we define a min-entropy order $P \leq_M P'$ by considering all possible a priori distributions, i.e.

$$P \leq_M P' \iff \forall \mu_h.\ H_\infty(h) - H_\infty(h|P) \leq H_\infty(h) - H_\infty(h|P').$$

Finally we define a *dissipativity* order $\leq_W$ based on security over all possible a priori distributions by

$$P \leq_W P' \iff \forall \mu_h.\ H(h|P) \leq H(h|P')$$

Proposition 8. *For deterministic programs the following relations hold:*

$$\forall P, P'.\ P \geq_W P' \iff P \leq_H P' \iff P \leq_M P'.$$

The first equivalence is intuitive and follows from the definition of W: the more information is leaked the less dissipation is required. The second equivalence is, in the light of differences between entropy and guessability, more surprising and is proved reasoning in terms of the observational equivalence in [25] or in more syntactic terms in [35].

6. Dynamics of Computation: timing channels

We now consider timing channels. As timing behaviours belong to the dynamics of the system, all considerations must be restricted to computers with a specific dynamics. Current computers are overwhelmingly clock-based. The problem with this choice of time evolution is that it is very wasteful: a state transition activated by a clock wastes far too much energy compared to the fundamental thermodynamic requirements and obfuscates the thermodynamic properties of timing channels arising from the correlation between minimum energy consumption and speed of computation.

Physicists interested in the thermodynamics of computation have studied low–energy yet powerful models of computation like Brownian computers [4, 14]. Biology is a good example of the reliability and complexity of such systems: an

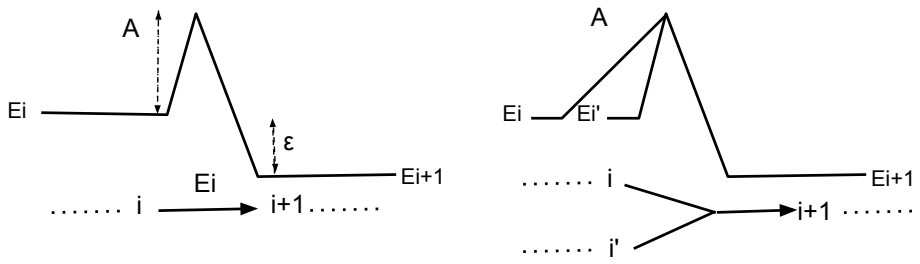


Figure 6: Transitions in a Brownian computer without (left) or with (right) branching. A: activation energy; E_i: energy of state i; ϵ: driving potential.

example of Brownian computing is given by DNA replication, transcription and translation.

In general, Brownian computers transverse a well-defined, deterministically arranged sequence of states under the influence of thermal agitation. Starting from the input state, the computer is allowed to advance or backtrack at random on the computational path until it reaches the final state of the computation (think of a complex chemical reaction proceeding backwards and forwards between intermediate products until final equilibrium). Mathematically Brownian computations are random walks, i.e. systems with forward and backward probabilistic transitions between states. Notice that there is no contradiction between random walk operation and the deterministic nature of the computation. The computer evolves along a well-defined sequence of states that it can visit; the randomness is about the forward/backward dynamics only and a trajectory may well have just one end point.

6.1. Evolution of Brownian computers

At a physical level a Brownian computer evolves by transversing a series of states separated by an energy barrier A comparable to the thermal agitation energy k_BT (Figure 6 left). It is a standard result in statistical mechanics that the probability of transition from one state to another, separated by a (positive) energy difference δE, is proportional to $\exp(-\delta E/k_BT)$ [14].

When the activation energy A becomes available to the system because of thermal fluctuations, the system has two alternatives: it can either proceed forwards with probability

$$f = C \exp\left(-(A - E_i)/k_BT\right) \tag{25}$$

or backtrack to the state E_i with probability

$$b = C \exp\left(-(A - E_{i+1})/k_BT\right), \tag{26}$$

where C is a normalisation constant.

Let $r = f/b$ be the ratio of the forward to backward transition rates. We can then write

$$k_BT \ln r = -(E_{i+1} - E_i) = -\delta E, \tag{27}$$

which shows that on the average the machine will move towards the state of lower energy. One can thus bias the computation forwards by setting $E_{i+1} < E_i$ by a small amount ϵ; this energy is effectively dissipated in the transition.

6.2. Branching

Before considering the evolution of the system in terms of computations, it is useful to extend our analysis to the case of the transition from two possible antecedent states to a single posterior state (Figure 6 right). We assume for simplicity that the two antecedent states have equal energy E_i.

In this case, we still have

$$f = C' \exp\left(-(A - E_i)/k_B T\right) \tag{28}$$

for the forward transition, but we should write

$$b = 2C' \exp\left(-(A - E_{i+1})/k_B T\right) \tag{29}$$

for the backwards transition to account for the fact that the system has two possibilities of moving backwards and only one of moving forwards (indeed, from the above if $E_i = E_{i+1}$ we obtain that $b = 2f$, which confirms that the choice is entirely unbiased).

From the above we obtain, for $r = f/b$, the expression

$$k_B T \ln(f/b) = k_B T \ln r = -k_B T \ln 2 - (E_{i+1} - E_i). \tag{30}$$

Remembering that $\delta\mathcal{S} = k_B \ln 2$ is the reduction in entropy associated to the transition from two possible antecedent states in step i to a single state in step $i + 1$ and assuming that T is held constant we can write this as

$$k_B T \ln r = -\delta E + T\delta\mathcal{S} = -\delta F \tag{31}$$

where F is the free energy of the system (more in general, when entropy variations are involved the transition probabilities are proportional to $\exp(-\delta F/k_B T)$).

Once again, computation will on the average go forward if the right hand side is larger than zero, and otherwise it will go backwards; however, this time, an entropy term that was absent from Equation 27 appears. This has interesting consequences for the kinetics of irreversible computations, that will be explored in the following sections.

6.3. Speed of computation

The link between the variation of free energy at each step and the speed of computation can be made more explicit for the case that ϵ is small [14]. Writing the forward rate as f and the backwards rate as b we have: $f = b + \theta$ and therefore:

$$\begin{aligned} -\delta F = k_B T \ln f/b = k_B T \ln(1 + \theta/b) \approx k_B T \theta/b = \\ = k_B T (f - b)/b \approx k_B T \frac{f - b}{(f + b)/2} \end{aligned} \tag{32}$$

where the linearisation on the first line and the approximation on the second hold in the limit of small θ. On the right hand side, $f - b$ is proportional to the velocity v_{drift} with which the computation moves forward. The denominator $(f + b)/2$ is the average rate of transition due to Brownian motion, and can be interpreted as the maximum speed at which the system would move forward, if all the transitions were in the right direction. We indicate this with v_{therm}. We can then write

$$\epsilon + T\delta\mathcal{S} = -\delta F \approx k_B T \frac{v_{drift}}{v_{therm}}, \tag{33}$$

where for convenience we have introduced the (positive) quantity $\epsilon = -\delta E$. Note that v_{therm} depends on the characteristics of the system and on the temperature rather than on the driving potential ϵ.

6.4. Irreversibility and dissipation

As Equation 33 shows, to a first approximation the velocity with which the computation proceeds depends linearly on ϵ. This energy is not stored in the system in any useful way, rather it is dissipated as heat at temperature T; it is the energy dissipation per step. The linear dependency is characteristic of diffusive processes [15].

The precise details of the energetic balance depend on the nature of the computation. For a logically reversible, one-to-one computation we can represent the succession of states that the computer transverses as a sequence with no bifurcations. In that case the variation of entropy of the computer is zero, i.e. $\delta\mathcal{S} = 0$ in equation 33. Hence for computation to proceed on the average it is enough that a small energy gradient ϵ is applied to bias the system towards forward evolution. In this case, the machine dissipates energy ϵ at each step; this can be seen as the price to pay for carrying out the computation at the desired speed. In the genetic apparatus, dissipation is of the order of $10^2 k_B T$ per operation [5]. However, if we are prepared to wait for the result ϵ can be arbitrarily small (ignoring random errors).

However, logically irreversible computations are a different case. Each time two computational paths converge the state space accessible to the system is reduced. This can be enough to make the drift velocity v_{drift} in Equation 33 negative unless the energy ϵ dissipated at each step is larger than $-T\delta\mathcal{S}$. For an elementary step of computation with two–way branching this is equivalent to the requirements of the Landauer principle. On the average, as shown in Section 4, when mediated across all possible inputs and their respective computational paths the minimum dissipation required is lower bounded by $W k_B T \ln 2$.

7. Time channels

The fact that a Brownian computer goes through a certain number of discrete states at a given average drift speed (equation 33), albeit with a slightly exotic dynamics, naturally implies the existence of time channels. Much as in the standard paradigm of synchronous, clock-driven computing, calculations requiring a higher number of steps will on the average take longer to complete.

For Brownian computers, however, the effectiveness of such time channels turns out to depend on the amount of energy used to drive the computation forward.

Surprisingly, Brownian computers also exhibit an entirely different, novel class of time channels that are intrinsically linked to the degree of irreversibility of a computation and that allow discriminating between computations requiring an identical number of operations.

In the following sections we will detail the two types of time channels and illustrate their dependency on energy and irreversibility.

7.1. Length of computation and time

The most obvious time channel is linked to the number of steps required by a computation. Let us for simplicity consider a one-to-one computation corresponding to a non-branching computational path, and assume that the energy profile for each of its steps is as displayed in Figure 6 (left) but with $\epsilon \gg k_B T$. Thus the activation energy is still low, but the large driving potential makes backward transitions very unlikely (in fact, irrespective of the level of branching of the computational path). In this case the system behaves more like a standard sequential machine than as a Brownian computer; the length of the computational path (together with the availability of the activation energy required for a transition) becomes the only variable determining the average duration of the computation. A similar behaviour can be expected for computational paths with little branching, where the entropy term in Equation 33 is small compared to the driving potential ϵ. We can think of this case as the "synchronous" limit.

As the driving potential decreases, however, the computer will spend more and more time retracing steps of the computation at random. In fact, without any driving force, a Brownian computer would drift away (in either direction) from the starting state by effect of thermal agitation. The variance of its position ν after $n = v_{therm}t$ random steps would be $\langle \nu^2 \rangle = n$. In a first approximation, we can assume this dispersion to be superposed to the net displacement represented by v_{drift}, thus making it more difficult for an attacker to estimate the number of steps of the computation from a timing measure. As the calculations in Section 7.3.1 show, for very low driving energies position uncertainty can effectively mask these timing channels.

7.2. Entropy timing channels

An entirely novel type of time channel directly related to the logical irreversibility of computation arises in Brownian computers when the driving potential is low. In this case, the contribution of the entropy term $T\delta\mathcal{S}$ in Equation 33 to v_{drift} cannot be discarded.

To see how this can induce a timing channel, consider the following example code:

```
l=0;
for(i =0; i< n; i++)
  { l=l+h[i];
   h[i]=0;}
```

```
l=0;
```

that computes the number of bits of the secret h that are set. The number of operations is here independent of the input — and so, in the standard setting, would be the time required by computation.

However, for Brownian computers equation 33 introduces a dependence of the speed of computation on input through the term $T\delta\mathcal{S}$. The key observation is here that the total entropy variation for the computation of $P(h)$ is $\Delta\mathcal{S} = -H(h|P = o)k_B \ln 2$, and that in general this will depend on the input. In other words, paths corresponding to different inputs will have different average degrees of branching per step (their length being equal).

Since it is reasonable to assume that the driving potential ϵ is set once for all independent of the input, this will necessarily lead to a different v_{drift} and therefore to a different average duration of the computation. Paths showing a higher degree of branching (maximum entropy reduction) will take longer, as the computer can be expected to drift backwards into the different possible antecedents of each node. Thus an attacker will be able to discriminate between inputs associated with a different degree of branching on the basis of the time taken by the computation. As calculation in Section 7.3.2 below show, this effect becomes more relevant for small driving potential, i.e. the situation in which "standard" timing channels are fuzzed by random oscillations.

7.3. Timing channels and variance

We present here a simple computational treatment of the two types of timing channels described in Section 7.1 and 7.2 above, that illustrates their main characteristic and the trade-offs involved.

In order to simplify calculations we assume that the computer can move forward with probability f and backwards with probability b at each and every step, and that each transition takes the same time.

We also assume that branching computational paths have a constant degree of branching at each node, as would be the case for example of a complete binary tree. In this case, since we are only interested in the level of the tree reached by the computer at a given time, we can still describe the computation as a one-dimensional asymmetric ($f \neq b$) random walk, where the current position corresponds to the level the system reached in the tree (see Figure 7).

Under these assumptions, the probability that the system is found in position (or at tree level) ν after a total of n transition is equal to the probability that exactly $(n + \nu)/2$ such transitions are forwards, and consequently $(n - \nu)/2$ of them are backwards:

$$B_f\left(\frac{n+\nu}{2}, n\right) = \binom{n}{\frac{n+\nu}{2}} f^{\frac{n+\nu}{2}} b^{\frac{n-\nu}{2}}. \tag{34}$$

where we assume that the binomial coefficient is zero if $(n + \nu)/2$ is not an integer or $|\nu| > n$.

If we assume that the last step of the computation is a potential well, so that the computer will get trapped when it reaches the end of the computation

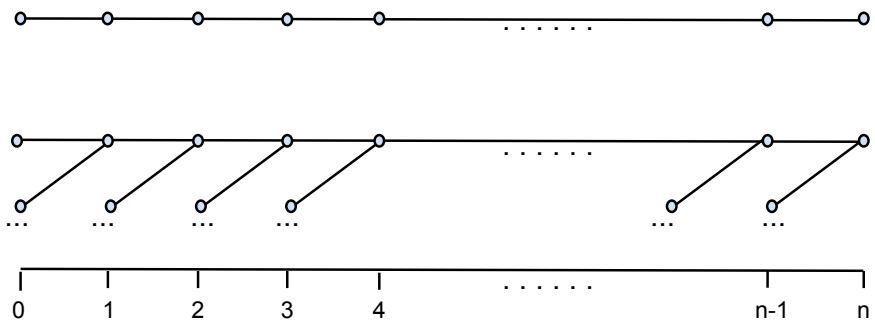


Figure 7: Two random walks (top: non-branching, bottom: uniformly branching) and the corresponding computational positions (tree levels)

for the first time (and some flag will be raised to signal that termination has occurred), it can be shown that the probability $\mu(n|\nu; f)$ that computation of ν steps will terminate in n transitions is given by

$$\mu(n|\nu; f) = \frac{\nu}{n} B_f\left(\frac{n+\nu}{2}, n\right) = \frac{\nu}{n}\binom{n}{\frac{n+\nu}{2}} f^{\frac{n+\nu}{2}} b^{\frac{n-\nu}{2}} \tag{35}$$

for positive ν.

7.3.1. Length of computation channel:

Let's consider two computations requiring a different number of operations $\nu_1 \neq \nu_2$. Let us assume that the driving potential ϵ is set so that the forward rate f is the same in the two cases.

According to the DeMoivre–Laplace theorem [33], for $|k - nf| \lesssim \sqrt{nfb}$ we have

$$\binom{n}{k} f^k b^{n-k} \simeq \frac{1}{\sqrt{2\pi nfb}} e^{-\frac{(k-nf)^2}{2nfb}}.$$

Substituting this into Equation 35 yields

$$\mu(n|\nu; f) \simeq \frac{\nu}{n} \frac{1}{\sqrt{2\pi nfb}} e^{-\frac{(\frac{1}{2}(n+\nu)-nf)^2}{2nfb}} \tag{36}$$

Notice that the graph of the last factor is not a Gaussian curve, as n also appears in the denominator of the exponent.

Figure 8 (left) shows the probability of the total number of transitions $\mu(n|\nu_1; f)$, $\mu(n|\nu_2; f)$ for ν_1 requiring 600 operations, ν_2 requiring 800 operations, and $f = 0.7$. Perhaps somewhat unsurprisingly we see that, for instance, a computation terminating around time 1150 is most likely to be ν_1, while one terminating after 1400 steps is more likely to be ν_2. This is akin of the common sense intuition about timing channels: longer computations take longer to terminate.

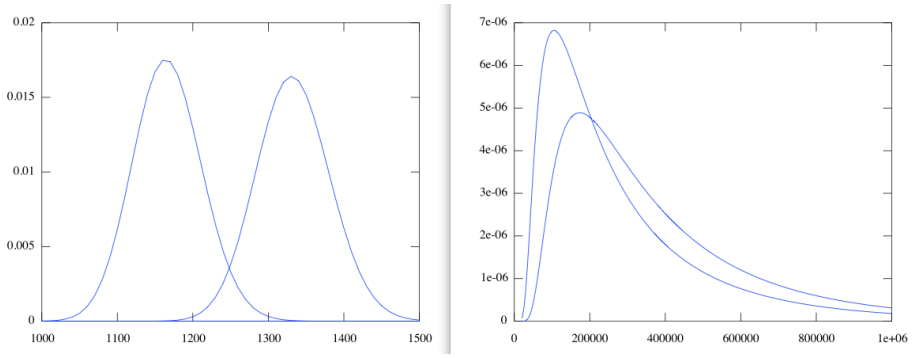


Figure 8: Halting probability for computations of different lengths and same forward rate as a function of the number of transitions. Left: the two curves are clearly separated when f is high. Here $\nu_1 = 600$, $\nu_2 = 800$ and $f = 0.7$. Right: the distributions overlap extensively when $f \gtrsim 0.5$ (here $f = 0.501$). Notice the very wide range of transition numbers over which this happens.

Interestingly however, as f decreases such channels tends to disappear as the variance of the distributions increases so as to blur the distinction between the two curves. Figure 8 (right) shows the distributions corresponding to the same computations for $f = 0.501$: over a huge range of total transition numbers there is no reliable way to determine which computation has finished, as both probabilities are comparable.

7.3.2. Entropy channels:

These timing channels are somewhat counterintuitive as they do not derive from a difference in the number of transitions but rather in the degree of branching, i.e. the difference in entropy variation between computations, that through Equation 33 results in a different speed. Consider two computations of the same length $\nu_1 = \nu_2 = \nu$ but different speed $f_1 \neq f_2$. To study these channels we consider the Log–likelihood of the probabilities:

$$\ln \frac{\mu(n|\nu; f_1)}{\mu(n|\nu; f_2)} = \ln \frac{f_1^{(n+\nu)/2} b_1^{(n-\nu)/2}}{f_2^{(n+\nu)/2} b_2^{(n-\nu)/2}} = \\ = \frac{1}{2}(n+\nu) \ln \frac{f_1}{f_2} + \frac{1}{2}(n-\nu) \ln \frac{b_1}{b_2} = \\ = \frac{1}{2} n \ln \frac{f_1 b_1}{f_2 b_2} + \frac{1}{2} \nu \ln \frac{f_1 b_2}{f_2 b_1} \quad (37)$$

If we now set $\phi = \ln(f_1/f_2)$, $\beta = \ln(b_1/b_2)$ we obtain

$$\ln \frac{\mu(n|\nu; f_1)}{\mu(n|\nu; f_2)} = \frac{1}{2} n(\phi + \beta) + \frac{1}{2} \nu(\phi - \beta). \quad (38)$$

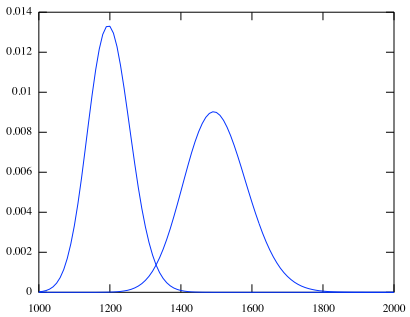


Figure 9: Timing channels due to branching. The distributions correspond to the halting probabilities for two computations of the same length (600 steps), the leftmost with $f = 0.75$ and the rightmost with $f = 0.70$.

Assuming that branching is higher on the second path than on the first path, so that $f_1 > f_2$, we have $\phi > 0$ and $\beta < 0$. Besides, it is easy to see that $|\phi| < |\beta|$. Therefore, the line in Equation 38 has negative slope and a positive intercept. For

$$n \leq \left\lfloor \frac{\beta - \phi}{\beta + \phi} \nu \right\rfloor = n^* \tag{39}$$

the computation is more likely to have followed the first path. Note that when f_1 is close to f_2 the slope of the line becomes small, making it more difficult to distinguish between the two paths (for $f_1 = f_2$ the likelihood ratio is identically 1). Conversely, for f_1 significantly larger than f_2 a small deviation from n^* is enough to tell apart the path with confidence.

Sample probability densities for this kind of timing channels (computed using the approximation in Equation 36) are illustrated in Figure 9.

Notice that length of computation and entropy channels will in general co-exist and that typical security counter-measures for timing channels like padding or noise insertion will need to take into account both kinds of channels.

8. Practical implications

The work here presented is of foundational nature and deals with general paradigms of computing rather than with specific implementations; its aim is to advance our scientific understanding of confidentiality. We make no claim at the moment about major applications of these ideas to come in the near future. It is however worth spending a few words in relating these ideas to some practical applications of thermodynamics to security.

The first that comes to mind is power analysis attacks. This kind of attacks, that rely on differential energy consumption in different circuit paths, have been very successful in breaking cryptographic implementations [34]; in fact they are among the most successful crypto security attacks to date [30]. For low energy devices such as Brownian computers these energy side channels could arguably

be studied using the energy-speed correlation from Sections 6 and 7. Other kind of power analysis, for example of authentication systems, could also in principle be related to the first part of this work, though accessing all information leaked by the system in specific states might require a more detailed modelling of the micro-states of the system based on statistical mechanics rather than on classical thermodynamics.

Overall key to the applicability of this work is the development of low energy computers. The energies we considered are minuscule as compared to the dissipation of nowadays transistors ($\approx 10^6 k_B T$ per transition). However nanotechnology is slowly lowering this figure to a point where they will no longer be irrelevant. Carbon nanotube memories with switching energies of the order of $10^3 k_B T$ have been feasible, at least at the prototype stage, for over a decade [27]. Very recently, experimental work implementing a Szilard engine [31] has brought the Landauer principle within the realm of experimental validation. Molecular computation has similarly proved its feasibility; early applications interestingly included cryptoanalysis [1] as well as autonomous DNA–based computation [3] quite related to the Brownian computing scenario we consider here. Timing aspects of DNA–based computers are actually the subject of current investigation [8]. For computers operating so close to reversibility the energy cost of security presented in this paper would clearly be significant. Once technology pushes devices to energy limits comparable to thermal agitation, further efficiency will only be achievable by making calculations reversible wherever possible. At that stage, security will become a hard lower bound on dissipation, and a secure system protecting a large amount of data will need to dissipate a comparatively sizeable amount of energy. Crucially if the dissipation of the system were below a reasonable multiple of W serious doubts on its security could be raised.

9. Conclusions

The study of thermodynamic aspects of computation dates back to the pioneers of computing starting with Von Neumann. Following works by Landauer and later Friedkin and Toffoli and Bennett illustrated how all computations can be executed reversibly. Thus dissipation, while of great practical importance, seems to have little foundational status in computer science.

Here we established a fundamental relation between dissipation and secure computation by proving that two of the main metrics of confidentiality in computer security are essentially measures of dissipation in the thermodynamic sense. These results provide thermodynamic foundations for confidentiality, with Landauer's principle thus implying a fundamental lower bound to the energetic cost of secure computation.

We also explored the relationship between the dynamics of computation and security for the case of Brownian computers. As we showed, this computing paradigm supports a novel type of time channels directly related to the irreversibility of computation, and thus to security. This further suggests that when

computer technology is pushed towards the limits of energy efficiency, interesting correlations between quantities normally considered unrelated by computer scientists (such as energy, entropy and time) may appear; these will require a detailed analysis.

Understanding the physics of confidentiality contributes to the debate on the role of irreversibility in other minimally dissipative systems including nano technologies, molecular and biological computation and quantum computing. Applied fields such as the study of power analysis attacks are also likely to benefit.

References

[1] L. M. Adleman, P. W. K. Rothemund, S. Roweis, E. Winfree: On applying molecular computation to the Data Encryption Standard, Journal of Computational Biology 6(1), 53–63, 1999

[2] M. S. Alvim, M. E. Andrés, K. Chatzikokolakis, C. Palamidessi: On the Relation between Differential Privacy and Quantitative Information Flow. ICALP (2) 2011: 60-76

[3] Y. Benenson, T. Paz-Elizur, R. Adar, E. Keinan, Z. Livneh, E. Shapiro: Programmable and autonomous computing machine made of biomolecules, Nature 414(6862):430–434, 2001

[4] C. Bennett. Logical Reversibility of computation. IBM J.Res.Develop. 17, 525-532. 1973.

[5] C. Bennett. Dissipation-error tradeoff in proofreading. Biosystems 11, pg 85–91. 1979.

[6] K. Chatzikokolakis, C. Palamidessi, P. Panangaden: Anonymity protocols as noisy channels. Information and Computation 206(2-4): 378-401 (2008)

[7] H. Chen, P. Malacaria: Quantifying maximal loss of anonymity in protocols. ASIACCS 2009: 206-217

[8] N. Aubert, Y. Rondelez, T. Fujii, M. Hagiya: Enforcing delays in DNA computing systems, 18th Intern. Conf. on DNA Computing and Molecular Programming (DNA18), Aarhus, Denmark, August 2012

[9] T. Chothia, V. Smirnov. A Traceability Attack against e-Passports. Financial Cryptography 2010: 20-34

[10] D. Clark, R. Hieron. Squeeziness: An information theoretic measure for avoiding fault masking. Information Processing Letters Volume 112, Issues 89, 30 April 2012, Pages 335-340

[11] D. Clark, S. Hunt, P. Malacaria: Quantitative information flow, relations and polymorphic types. Journal of Logic and Computation, 18(2):181-199, 2005.

[12] D. Clark, S. Hunt, P. Malacaria: A static analysis for quantifying information flow in a simple imperative language. Journal of Computer Security, Volume 15, Number 3. 2007.

[13] T. Cover, J. Thomas. Elements of Information Theory. Wiley-Interscience publications. 1991.

[14] R. P. Feynman. Feynman Lectures on Computation. Edited by A. Hey and R. Allen. Addison Wesley 1996.

[15] R. P. Feynman, R. B. Leighton and M. Sands. The Feynman Lectures on Physics. Vol 1. Addison Wesley 1964.

[16] E. Fredkin, T. Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21:219253, 1982.

[17] B. Köpf, D. Basin: An information-theoretic model for adaptive side-channel attacks. Proceedings ACM conference on Computer and Communications Security, 2007, 286-296.

[18] B. Köpf, G. Smith: Vulnerability Bounds and Leakage Resilience of Blinded Cryptography under Timing Attacks. CSF 2010: 44-56

[19] K. Chatzikokolakis, C. Palamidessi, P. Panangaden: On the Bayes risk in information-hiding protocols. Journal of Computer Security (JCS) 16(5):531-571 (2008)

[20] J. Heusser, P. Malacaria: Quantifying Information Leaks In Software. Proceedings ACM Annual Computer Security Applications Conference, ACSAC 2010, Austin, Texas. ACM 2010.

[21] R. Landauer. Dissipation and heat generation in the computing process. IBM J.Res.Develop., 5, 148-156. 1961.

[22] H. Leff and A. Rex editors. Maxwell's Demon 2, Entropy, Classical and Quantum Information, Computing. Institute of Physics publishing 2003.

[23] P. Malacaria and F. Smeraldi, The Thermodynamics of confidentiality, in Proc. of IEEE Computer Security Foundations Symposium, CSF 2012

[24] P. Malacaria. Assessing security threats of looping constructs. Proc. ACM Symposium on Principles of Programming Language, POPL 2007.

[25] P. Malacaria. Algebraic foundations for quantitative information flow, Mathematical Structures in Computer Science, in press.

[26] A. Rényi: On measures of information and entropy. Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability 1960: 547-561.

[27] T. Rueckes, K. Kim, E. Joselevich, G. Y. Tseng, C.-L. Cheung, C. M. Lieber: Carbon nanotube–based nonvolatile random access memory for molecular computing, Science Vol. 289, July 7th, 2000: 94-97

[28] G. Smith: On the Foundations of Quantitative Information Flow. In Proc. FOSSACS 2009: Twelfth International Conference on Foundations of Software Science and Computation Structures LNCS 5504, pp. 288-302, York, UK, March 2009.

[29] G. Smith: Quantifying Information Flow Using Min-Entropy. In Proc. QEST 2011: 159-167

[30] F.-X. Standaert, N. Veyrat-Charvillon, E. Oswald, B. Gierlichs, M. Medwed, M. Kasper and S. Mangard. The World Is Not Enough: Another Look on Second-Order DPA. In: Advances in Cryptology - ASIACRYPT 2010, pages 112-129. Springer LNCS 6477, December 2010.

[31] S. Toyabe, T. Sagawa, M. Ueda, E. Muneyuki, M. Sano (2010-09-29), Information heat engine: converting information to energy by feedback control, Nature Physics 6 (12): 988-992, http://dx.doi.org/10.1038/nphys1821, arXiv:1009.5287, Bibcode 2011NatPh...6..988T.

[32] K. Zhang, Z. Li, R Wang, X. Wang, and S. Chen. Sidebuster: Automated Detection and Quantification of Side-Channel Leaks in Web Application Development. In Proc ACM CCS 2010.

[33] A. Papoulis, S. U. Pillai: Probabilities, random variables and stochastic processes, McGraw-Hill 2002

[34] P. Kocher, J. Jaffe, B. Jun. Differential Power Analysis. in Advances in Cryptology - Crypto 99 Proceedings, Lecture Notes In Computer Science Vol. 1666, M. Wiener, ed., Springer-Verlag, 1999.

[35] H. Yasuoka, T. Terauchi: Quantitative Information Flow - Verification Hardness and Possibilities. CSF 2010: 15-27