The Thermodynamics of Confidentiality

Pasquale Malacaria, Fabrizio Smeraldi
School of Electronic Engineering and Computer Science
Queen Mary University of London
London, UK
{pm,fabri}@eecs.qmul.ac.uk

***Abstract*—This work, of a foundational nature, establishes a connection between secure computation and the 2nd principle of thermodynamics. In particular 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 remaining uncertainty as defined in Quantitative Information Flow, K_B the Boltzmann constant and T the system temperature.**

By contrast, for probabilistic computations thermodynamic work can be extracted from secure systems: in this case, again using information theoretic results, we provide bounds on the amount of work that can be extracted.

Further we show that in deterministic systems the dissipated energy is an upper bound on Smith's remaining vulnerability; by doing so we provide the first thermodynamic interpretation of guessability.

Crucially, unlike much literature on the physics of computation, our focus is not a universal model but a software field of great practical relevance, namely security. We see this work as a genuine scientific advance with the potential to enhance the understanding of both confidentiality and dissipative systems in physics.

***Keywords*-Security, Information Theory, Thermodynamics, Quantitative Information Flow.**

I. Introduction

Data protection seems far from the concrete, bolt–and–chain world of "real" security. In the real word to guarantee higher security you fire up the forge to weld a stronger, heavier chain. Nobody probably would dream of heating a computer to protect data, but this, we aim to show here, is what secure algorithms actually do. Increasingly, low–consumption devices suggest that computation has no intrinsic energy requirements: if our processors were efficient enough, theoretically as Bennett proved [2] we would get it for free. This is true for most calculations, but false for secure computation. As we show, any secure algorithm needs to generate heat: indeed, the greater the security, the more energy is dissipated in "making" it — quite like forging the chain. While the energy involved is minuscule compared with the inefficiencies of nowadays computers, still data security means heat — it means fuel — and confidentiality is, to some extent, a branch of thermodynamics.

Early definitions of confidentiality were very restrictive: 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).

Motivated by the unavoidability of leakage Quantitative Information Flow [7], [8] 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 that evaluated on these probabilities allows the inference 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 (we assume the secret not to be an observable output). 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 has zero entropy [7].

Quantitative Information Flow has been applied among others to side channel attacks analysis [14], [15], [5], to measure confidentiality leaks in the Linux Kernel [18], to database security analysis [1], to analysis of anonymity protocols [3], [4], to side channels leaks in web applications [29] and avoidance of fault masking [6].

A. Contributions

The overall contribution of this paper can be seen as laying down in a precise sense the thermodynamic foundations of confidentiality.

In Section IV we consider an idealised physical model that is commonly used in the literature of thermodynamics of computation [12], [2]. We generalize it to model an arbitrary number of states with a non-uniform distribution, thus providing a conceptual model for the most general statement of the Landauer principle. We then show that the notion of remaining uncertainty W from Quantitative Information Flow is, up to the multiplicative factor $K_B T \ln 2$, precisely the minimum dissipation associated to secure computation (equation 17 and proposition 3).

Section IV-D 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 20), although erasure remains an irreversible operation. Again the energy is bounded by the quantity W which is in this case non-positive.

Section V investigates the thermodynamics of Smith's notion of vulnerability and proves that remaining vulnerability is in general a lower bound on W (proposition 6). The bound becomes an equality if and only if the remaining vulnerability 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 V-A.

II. Background

We just recall here a few information theoretical notations and definitions we will use in the paper.

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

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

where $\log$ is logarithm in base 2.

The definition extends to random variables. The entropy of a random variable X is

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

Given two random variables X, Y the conditional entropy $H(X|Y)$ is defined by $H((X,Y)) - H(Y)$ where (X,Y) is the joint random variable X, Y. It is a measure of 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, how much information they share.

A. 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 [7]:

$$I(h; P|l) = H(h|l) - H(h|P, l) \tag{1}$$

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 (1) reduces to mutual information $I(h; P)$ and the following holds:

$$\begin{aligned} I(h; P) &= H(h) - H(h|P) && (2) \\ &= H(P) - H(P|h) && (3) \\ &= H(P) && (4) \end{aligned}$$

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 [21] (resp. [18]) 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 2, 3 and 4.

In its formulation $W = H(h|P)$ the quantity W is known in the Quantitative Information Flow community as *security* or *remaining uncertainty* [21], [26]. 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 IV-D.

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 17). In other words $W \ln(2) K_B T$ is the minimum dissipation any system implementing that program has to emit 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 [21], [18].

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 of Bennett [2].

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.

III. The thermodynamics of computation

Thermodynamic aspects of computation have been considered, among others, in the works of Bennett [2], Feynman [12], Landauer [19]. Computation is carried out by physical systems, that are for all purposes governed by the laws of physics. Thus it makes sense, for instance, to ask how much energy is required to perform a certain computation. As it turns out, this question is best answered in statistical terms, starting from a correspondence between a logical state of the computer and the physical state of the underlying machinery, that naturally becomes a correspondence between the entropy of the physical system and the concept of entropy in information theory.

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.

A. 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, perfect gases of one molecule, 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 [13], [12], with some qualifications that will become clear below.

B. 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 implies that all logically reversible functions can be computed reversibly without energy expenditure. More practical models of reversible computation include the universal reversible controlled-XOR gate introduced by Friedkin and Toffoli [13]. 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 desireable to do so.

C. 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:

$$\Delta\mathcal{S} \geq \frac{\delta Q}{T} \tag{5}$$

where Q 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 an equilibrium with a single heat source at temperature T (the environment), Equation 5 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:

$$0 = \Delta\mathcal{S} > \oint \frac{\delta Q}{T} = \frac{1}{T}\oint \delta Q = \frac{\Delta Q}{T} \tag{6}$$

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, ie of a complete specification of all its degrees of freedom, entropy can be written as

$$\mathcal{S} = -K_B \sum_i p_i \log p_i, \tag{7}$$

which bears a striking analogy to the Shannon entropy H. Indeed, since logical states in a computation are in one-to-one correspondence with the physical states of the computer, Equation 5 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.

D. From Maxwell's demon to the Landauer principle

Another consequence of Equation 5 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 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. In a simplified version of the argument, the molecule is just one, and the demon is able to trap it into one of the two chambers at its will. This chamber can then be expanded by letting the particle do work against the partition, thus extracting energy $K_BT\ln 2$. 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 focussing 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 [20]. 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 $K_BT\ln 2$. That this is in general the minimum cost for the cancellation of one bit of information has become enshrined in the so–called Landauer principle [19]; this principle has very recently also been experimentally demonstrated [28]. As argued by Bennett [2], Feynman [12], this intrinsic cost of cancelling information is the key consideration in the thermodynamics of computation.

IV. Physical model of secure computation

A. A simple two state register

In this section, we will derive a few basic results on the energetic cost of erasing information from 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 specific 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 (quite like the case of Maxwell's demon, but without the trap door and its demonic operator). 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

$$\int_2^1 \mathcal{P}dV = K_BT\int_2^1 \frac{1}{V}dV = -(\ln 2)K_BT \tag{8}$$

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. Within this set of boxes, the molecule is in the left half of the box in proportion μ_1 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

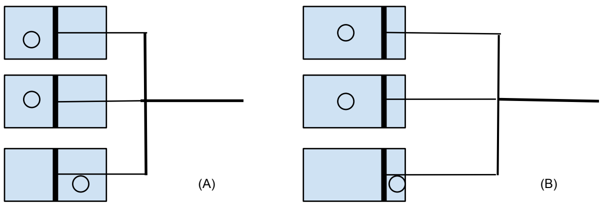

Figure 1. (A) the system before expansion, (B) system after expansion

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 1 illustrates the idea for a system consisting of a two states with probabilities $\frac{1}{3}, \frac{2}{3}$

Again assuming that each chamber initially has unit volume, and averaging across the ensemble, we have $\mathcal{P}_i = \mu_i K_B T$, 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.

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

$$\begin{aligned}
\frac{W_{exp}}{K_BT} = \frac{1}{K_BT}\int_1^{2\mu_1}(\mathcal{P}_1 - \mathcal{P}_2)dV = \\
= \int_1^{2\mu_1}\frac{\mu_1}{V}dV - \int_1^{2\mu_1}\frac{\mu_2}{2-V}dV = \\
= \mu_1 \ln 2\mu_1 - \mu_2 \ln\frac{2-1}{2-2\mu_1} = \\
= \mu_1 \ln 2\mu_1 + \mu_2 \ln 2\mu_2 = \\
= \mu_1 \ln 2 + \mu_1 \ln \mu_1 + \mu_2 \ln 2 + \mu_2 \ln \mu_2 = \\
= \ln 2 - H(\mu_1, \mu_2) \ln 2 \qquad (9)
\end{aligned}$$

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 8 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

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

B. 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 shaft that can be actuated independently. Figure 2 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

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

$n-1$ being the volume.

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

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

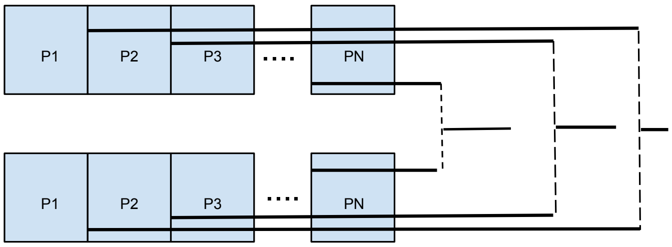

Figure 2. Modelling a system with N states

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.

Thus work done during the expansion, similarly to equation 9 is

$$\frac{W_n}{K_BT} = \frac{1}{K_BT}\int_{n-1}^{n(1-\mu_n/M_n)} (\mathcal{P}_{n-1} - \mathcal{P}_n)dV =$$
$$= \int_{n-1}^{n(1-\mu_n/M_n)} \left(\frac{M_{n-1}}{V} - \frac{\mu_n}{n-V}\right) dV =$$
$$= M_{n-1}\ln\frac{n(1-\mu_n/M_n)}{n-1} + \mu_n \ln\frac{n\mu_n}{M_n} =$$
$$= M_{n-1}\ln\left(\frac{n}{n-1}\frac{M_{n-1}}{M_n}\right) + \mu_n \ln\frac{n\mu_n}{M_n} =$$
$$= M_n \ln\frac{n}{M_n} - M_{n-1}\ln\frac{n-1}{M_{n-1}} + \mu_n\ln\mu_n \quad (10)$$

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:

$$\frac{W_{exp}}{K_BT} = \sum_{n=1}^{N}\frac{W_n}{K_BT} =$$
$$= \sum_{n=1}^{N}\left(M_n\ln\frac{n}{M_n} - M_{n-1}\ln\frac{n-1}{M_{n-1}}\right) +$$
$$+ \sum_{n=1}^{N}\mu_n\ln\mu_n. \quad (11)$$

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

$$\frac{W_{exp}}{K_BT} = M_N\ln\frac{N}{M_N} + \sum_{n=1}^{N}\mu_n\ln\mu_n =$$
$$= \ln N + \sum_{n=1}^{N}\mu_n\ln\mu_n. \quad (12)$$

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:

$$\frac{W_{comp}}{K_BT} = \int_N^1 \frac{1}{V}dV = \ln N \quad (13)$$

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

$$W_{reset} = W_{comp} - W_{exp} \quad = \quad (14)$$
$$= -\sum_{n=1}^{N}\mu_n\ln\mu_n K_BT \quad = \quad (15)$$
$$= H(\mu_1 \ldots, \mu_N)K_BT(\ln 2) \quad (16)$$

C. Universal factorisation of secure computations

Bennett [2], Friedkin and Toffoli [13] 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:

$$\begin{tikzcd}
(h,0) \arrow[r, "R"] \arrow[dr, dashed, "\tilde{P}"'] & (S(h),P(h)) \arrow[d, "\Pi"] \\
 & (0,P(h))
\end{tikzcd}$$

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 registers $S(h)$ hold 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 3 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 3 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. :

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

Combining this with equation 14 it then follows that the cost of resetting S, averaged over all program outputs, is

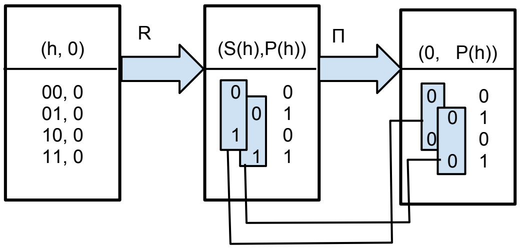

Figure 3. Secure computation of $\texttt{l} = \texttt{h}\%2$ on a two bit secret

$$\begin{aligned}\left(\sum_k \mu(P_k) \sum_{h\in P^{-1}(P_k)} \mu(h|P_k)\ln\frac{1}{\mu(h|P_k)}\right) K_B T &= \\ = H(h|P)K_BT\ln 2 &= \\ = WK_BT\ln 2 & \qquad (17)\end{aligned}$$

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

This result is universal i.e.

Proposition 3: $WK_BT\ln 2$ 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 $WK_BT\ln 2$, then $\tilde{P}_0 \circ R^{-1}$ is effectively an implementation of the reset operation Π that violates Landauer's principle.

D. 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; such cancellation by randomization is called erasure. Considering the graph in Section IV-C, we replace the reset operator Π with an erasure operator E:

$$(S(h), P(h)) \overset{E}{\longmapsto} (\epsilon, P(h)) \qquad (18)$$

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) $S(h)$ 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

$$H(\tilde{P}_e) = H(P) + \log(|S(h)|) \qquad (19)$$

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

$$\begin{aligned} W &= H(h) - H(\tilde{P}_e) = \\ &= H(h) - H(P) - \log(|S(h)|) \leq 0 \qquad (20)\end{aligned}$$

The second equality is illustrated by the commutative diagram in Figure 4: we know from section IV-C that Π has cost $H(h) - H(P)$ and Π' has, by Landauer principle, cost $\log(|S(h)|)$. Inequality 20 then follows because $S(h)$ and $P(h)$ 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 19 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 1); 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 $W = H(h) - H(P)$ instead of $W = H(h|P)$. While the two definitions are equivalent in the deterministic setting they differ in the probabilistic one. In fact by choosing $W = H(h|P)$ then, as the conditional entropy is always non-negative we would conclude that dissipation to protect confidential data is needed 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 state (an alternative argument is provided by using Bennett's fuel value [12], [2]) and this work can exceed the work needed to protect confidential data hence $H(h|P)$ would be an imprecise definition.

V. The thermodynamics of vulnerability

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, notably the concept of vulnerability recently put forward by Geoffrey Smith [26]. As we here show, vulnerability 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.

Vulnerability quantifies the loss of confidentiality in terms of the difference between the probability of guessing the secret before and after observing the output of a program.

$$\begin{tikzcd}
(h,0) \arrow[r, "R"] \arrow[rr, dashed, bend left, "\tilde{P}_e"] \arrow[dr, dashed, "\tilde{P}"'] & (S(h),P(h)) \arrow[r, "E"] \arrow[d, "\Pi"] & (\epsilon,P(h)) \arrow[dl, "\Pi'"] \\
& (0,P(h)) &
\end{tikzcd}$$

Figure 4. Relation between erasure and resetting

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 called *vulnerability*, based on the Renyi min-entropy [23]. The vulnerability $V(P)$ of a program is the difference between its a priori guessability in one try: $G(h) = -\log(\max_{h_i \in h} \mu(h_i))$ and the a posteriory guessability in one try $G(h|P)$, expressed as a min-entropy conditioned over all possible values of the observables:

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

The a posteriori conditional guessability $G(h|P)$ is the log of the complement of the Bayes risk [16] and is also called *remaining vulnerability*. In the case of our examples, the vulnerability 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 the remaining vulnerability $G(h|P)$. If the remaining uncertainty W is the minimum dissipation what, if anything, is remaining vulnerability in thermodynamic terms?

A clear connection between remaining vulnerability and thermodynamics is given by the following result:

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

$$G(h|P) = \log(|h|) - \log(|P|).$$

The proof of the above is a consequence of the following facts:

1) The channel capacity of the two measures coincides, i.e.
$$\max_{\mu(h)}(G(h) - G(h|P)) = \max_{\mu(h)}(H(h) - H(h|P))$$
2) the channel capacity for vulnerability is given by the uniform distribution on the input h
3) the channel capacity for Shannon Leakage is given by the uniform distribution on the outputs of the program
4) in both cases the channel capacity is the log of the number of outputs of the program (noted $\log(|P|)$).

The thermodynamic interpretation of $G(h|P)$ 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) $G(h|P) = W$
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 $2^{|h|}$.

Notice also that by known results [26] relating channel capacity with vulnerability with h uniformly distributed proposition 5 also characterise dissipation associated to an implementation of the program where the leakage is maximal.

If we relax the condition about the input being uniformly distributed then remaining vulnerability always underestimate the dissipation i.e.

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

$$G(h|P) \leq W.$$

We prove that

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

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

$$\sum_{ij} h_{ij} \log h_{ij} \leq \sum_{ij} h_{ij} \log h_i^\star = \sum_i b_i \log h_i^\star \tag{21}$$

so that it will suffice to prove

$$\begin{aligned} \sum_i b_i \log b_i - \sum_i b_i \log h_i^\star + \log \sum_i h_i^\star \quad &= \\ = \sum_i b_i \log \frac{b_i}{h_i^\star} + \log \sum_i h_i^\star \quad &\geq 0 \end{aligned}$$

From Theorem 2.7.1 in Cover and Thomas [10] we have that

$$\sum_i b_i \log \frac{b_i}{h_i^\star} \geq \sum_i b_i \log \frac{\sum_i b_i}{\sum_i h_i^\star} \tag{22}$$

Replacing the first term in 22 with the above we have

$$\begin{aligned} \sum_i b_i \log \frac{\sum_i b_i}{\sum_i h_i^\star} + \log \sum_i h_i^\star \quad &= \\ = \log \frac{1}{\sum_i h_i^\star} + \log \sum_i h_i^\star \quad &= 0 \end{aligned}$$

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

We can now strengthen proposition 5 to precisely characterise when dissipation and remaining vulnerability coincide:

Proposition 7: $G(h|P) = W$ 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 21 is strict. If we relax the requirement of uniform distribution on the output then inequality 22 is strict. Any of these will make proposition 6 a strict inequality.

A. 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 formalize this by

$$\forall \mu_h . H_{\mu_h}(P) \leq H_{\mu_h}(P').$$

where μ_h ranges over all distributions on h.

Similarly we define a vulnerability order $\leq_V$ by considering vulnerability over all possible a priori distributions, i.e.

$$\forall \mu_h . V_{\mu_h}(P) \leq V_{\mu_h}(P').$$

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

$$\forall \mu_h . W_{\mu_h}(P) \leq W_{\mu_h}(P')$$

where $W_{\mu_h}(P) = H_{\mu_h}(h) - H_{\mu_h}(P)$.

Proposition 8: For deterministic programs the following relations hold:

$$\forall P, P'. \;\; P \geq_W P' \iff P \leq_H P' \iff P \leq_V P'.$$

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 [22] or in more syntactic terms in [31].

VI. Practical implications

The work here presented is of foundational nature and relying on ideal physical models; 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 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 of energy consumption in different paths in the circuits, have been very successful in breaking cryptographic implementations [30]; in fact they are among the most successful crypto security attacks to date [27]. However they do not directly relate with the work here presented. The reason is that, given a particular key, by definition encryption is a reversible computation, hence can in our ideal physical model carried on with no dissipation. Hence current crypto power analysis attacks are due to inefficiency of modern technology and not to fundamental physical laws. Other kind of power analysis, for example of authentication systems, could be in principle related to this work, though accessing all information leaked by the system in specific states might require a more detailed modelling of the microstates of the system based on statistical mechanics rather than on classical thermodynamics.

While the energies involved are minuscule as compared to the dissipation of nowadays transistors ($\approx 10^8 K_B T$ per transition), 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 [24]. More recently, experimental work implementing a Szilard engine [28] has brought Landauer principle within the realm of experimental validation. 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.

VII. Future directions

In this work we have focused on the state of the input and output registers and energetic aspects using quasi-static reversible transformations, and as such we have not directly treated side channels like time. Several applications of Quantitative Information Flow focus on side channels, in particular time channels. Modelling these observations from a thermodynamic point of view would require enlarging the boundary of the system, for instance by adding time as a register to the system in order to support the analysis of timing channels; while some of the analysis could be carried out by considering the time register as part of the history register, non-equilibrium aspects of the evolution of the system may become relevant. Ultimately, one should consider a larger system of which both the computer and the observer are part. This would be reminiscent of the Maxwell demon where the memory of the demon is considered part of the system. A better understanding of the thermodynamics of side channels will require further work. On a similar theme recent work [17] outlines a very interesting abstract model of reversible computation based on type isomorphism that parallels the relation between open and closed physical systems. The authors of that work rightly notice connections with Quantitative Information Flow; in those terms our work could be seen as complementing [17] by providing a fine grained description of the physical and information theoretical aspect of the security "information effect".

Also we focused on two metrics but it would be very interesting to study possible thermodynamic properties of other metrics of confidentiality like differential privacy [11] and beliefs metrics [9].

Finally the role of Landauer principle in quantum computing is not clear [25]; it would be hence interesting to investigate the information theoretical and physical foundations of confidentiality in that field.

VIII. 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, namely information leakage and vulnerability, 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. Understanding the physics of confidentiality contributes to the debate on the role of irreversibility in other minimally dissipative systems such as 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] 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

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

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

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

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

[6] D. Clark, R. Hieron. Squeeziness: An information theoretic measure for avoiding fault masking. Information Processing Letters Volume 112, Issues 89, 30 April 2012, Pages 335340

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

[8] 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.

[9] M. Clarkson, A. Myers, F. Schneider: Belief in Information Flow. CSFW 2005: 31-45

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

[11] C. Dwork. Differential Privacy. In Proc. International Colloquium on Automata, Languages and Programming (ICALP) 2006, p. 112.

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

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

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

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

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

[17] R. James, A. Sabry. Information Effects. In proceedings POPL 2012, ACM 2012.

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

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

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

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

[22] P. Malacaria. Algebraic Foundations for Information Theoretical, Probabilistic and Guessability measures of Information Flow CoRR abs/1101.3453: (2011)

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

[24] 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

[25] L. del Rio, J. Aberg, R. Renner, O. Dahlsten,V. Vedral The thermodynamic meaning of negative entropy. Nature 474, 6163 (02 June 2011) doi:10.1038/nature10123

[26] 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.

[27] 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.

[28] 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): 988992. arXiv:1009.5287. Bibcode 2011NatPh...6..988T. doi:10.1038/nphys1821.

[29] 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.

[30] 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.

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