From Qualitative to Quantitative Information Erasure

Adedayo O. Adetoye and Michael H. Goldsmith

Cyber Security Centre
Department of Computer Science, University of Oxford
Oxford, United Kingdom
Emails: adedayo.adetoye@cs.ox.ac.uk, michael.goldsmith@cs.ox.ac.uk
http://www.cybersecurity.ox.ac.uk/

Abstract. We define a quantitative measure of information erasure as a dual of the well-understood notion of quantitative information release. Our journey begins from a qualitative, equivalence relations-based, definition of information erasure and release, which we show to be tightly linked to the quantitative measures of these notions. In particular, given the necessary probability distribution over the inputs of a deterministic system, we show that the quantitative measures of erasure and release are directly derivable from the equivalence relations-based definitions. However, we observe that the quantitative definitions, unlike the qualitative ones, are less expressive and may suffer from practical problems such as *erasure* and *release occlusion* – a problem, which at its core is attributable to the symmetry of the information-theoretic entropy definition.

1 Information Erasure and Release

There is often a need to erase information in real systems. In particular, a system that processes confidential data may be expected to remove pieces of sensitive information from the body of information that it propagates. For example, statistical databases may not propagate sensitive information, which must be erased; but the database must release sufficient non-sensitive information to be useful for statistical purposes. A more everyday example requiring information erasure is e-commerce, where various pieces of data on a credit card used must not be stored by the merchant. The Payment Card Industry [18], which specifies standards for payment processing, stipulates which data must not be retained by a merchant, even though the data may be required to complete a transaction. For example, the *card verification code*, which is used to prevent card-not-present frauds, must not be stored by the merchant [18]. There are also restrictions on the display of the *primary account number* (PAN) on screens or receipts, e.g. the *first six* and the *last four digits* are the *maximum* allowed to be displayed – the other digits must be masked (erased).

Note that in these examples, as with other situations where information erasure is desired, erasure often goes hand-in-hand with information release: e.g. some PAN digits may be released whereas others must be erased. So, it is reasonable to study erasure in the context of information release. It is even better if the two can be accommodated under a single uniform policy model, as we propose. As a general observation, it is desirable to be able describe security requirements as an extensional policy statement independently of the operational properties or implementation of the system that satisfies

the requirement. This separation of concerns is a well-understood good design principle of allowing policies and systems to be separately developed. A verification mechanism then ensures that the implementation conforms to the desired policy. The policy model proposed in this paper is extensional and describes the information security requirements directly as constraints on information release and erasure independently of an enforcement mechanism.

1.1 Partial Erasure Requirements

We consider *partial erasure* requirements as a generalisation of erasure policies and a dual of partial information-release policies. In a technical sense that will be made clear by Corollary 1 to Theorem 1, partial erasure goes in tandem with partial release, and the totality of one excludes the other for a given unit of information. To illustrate this observation, consider a credit card PAN made up of 16 digits, where we intend to completely erase the first 12 and release the last four. We may thus view the PAN as being made up of two conceptual units of information or fields: the first 12 digits and the last four digits. Even when we completely delete the first field, it is a total erasure of that field, but a partial erasure of the whole body of information since we release the second field. However, we may also have partial erasure of a unit of information, e.g. we may wish to erase all, but the parity of the first field. Total erasure, where we delete all the information, is merely a special case of information erasure that is symmetric to the information-flow notion of *noninterference* [10] because they both prevent or disallow any information release.

Now suppose that the 16-digit credit card PAN above is a pair taken from the set $X_1 \times X_2$, where $X_1 = \{n_1 n_2 \dots n_{12} \mid \forall i.n_i \in D\}$, $X_2 = \{n_{13} n_{14} n_{15} n_{16} \mid \forall i.n_i \in D\}$, and $D = \{0, 1, \dots, 9\}$; and that the payment processing application, which must erase the first 12 digits is modelled by a function $f : X_1 \times X_2 \to X_2$ defined as $f((x_1, x_2)) = x_2$. The function erases or suppresses information about X_1 inputs from the output so that an observer which sees only the output x_2 (say, on the receipt generated by the payment processing application) cannot learn anything about x_1 (if it is independent of x_2), or at best can only learn as much as can be inferred from x_2. In the second scenario, where we wish to erase all but the parity information of the first 12 digits, one may imagine a payment processor whose output is modelled by the function $g((x_1, x_2)) = (x_1 \bmod 2, x_2)$, which does that.

This example already alludes to some basic assumptions that we make concerning information erasing systems and the attack model, where we represent systems by functions. They are the following:

1. The attacker can observe the function result (corresponding to the relevant system output), but not (necessarily[1]) the input. Since our primary concern in this paper is to characterise *what* information is erased and/or released, we do not care about from *whom* or *where* the information comes from. Of course the *who* and *where* dimensions matter in specific applications to information security and integrity, but

[1] Note that, if required, the model can capture situations where the attacker contributes to the input or can observe portions of the inputs.

these are orthogonal concerns to the characterisation of *what* is erased or released: the same degree of input erasure may be judged bad in the context of integrity, but good in the context of secure release, for example.
2. The attacker knows the system model, algorithm(s) employed, or, conceptually, the function $f, g, \ldots$. This is a reasonable assumption for systems that implement standard algorithms and protocols (the majority of standard security-sensitive software systems), or software that might have been potentially implemented or tampered with by a malicious author (as in Internet-downloaded software). In other words, we do not factor in system obscurity as a defence mechanism within the model.

Our goal is to characterise the degree of erasure that a system achieves over its input domain X (which may or may not be structured) given some functional model $f : X \to Y$ of how the system transforms its inputs to relevant outputs in the domain Y.

2 Modelling Information Erasure in Computational Systems

We consider deterministic computational systems which process inputs from some input set X and produce outputs in some set Y. Being deterministic, we may model such computational systems as functions of the form $f : X \to Y$, such that for any given input $x \in X$ supplied to the system modelled by f, the observed output of the system is $f(x) \in Y$. Whenever this property holds for a function f and a system, we say that the system is *modelled by* f. Note that the functional model only relies on the determinism of the system, and may be applied to both *batch-processing* and *interactive* computational systems. The difference between the two being that, as opposed to batch-processing systems, input and output may be interleaved in a single execution of an interactive system. Interaction between the system and its environment may have consequences for information erasure/flow because it provides an opportunity for the environment to revise its behaviour based on the output observed so far. However, due to determinism, any particular strategy of the system and its environment may be encoded as an input sequence $x \in X$, which will always produce the same observation $f(x) \in Y$ on any run. Thus, interaction, in particular, does not affect our definitions and analyses.

Now, when an output $y \in Y$ is observed in a system modelled by $f : X \to Y$, this tells us that the input must have been taken from the set $f^{-1}(y)$, and any $x, x' \in f^{-1}(y)$ may have produced the observed output y. Without further information, there is no justification for preferring x over x' as the actual input that produced the output y. But we can confidently rule out any $x'' \in X$ as a possible input when we observe the output y and $x'' \notin f^{-1}(y)$. We thus gain information about inputs by being able to *distinguish* between possible inputs and those that are not, based on the observed output. We say that, based only on the observation of the output y, x is *indistinguishable* from x' as the actual input that produced the output y whenever $x, x' \in f^{-1}(y)$. Alternatively, we say that the system modelled by f *erases* the distinction between x and x'. This gives us a way to qualitatively characterise the information released/erased by a system modelled by f, namely, through the *kernel* κ_f of the function f, which is an equivalence relation over the input space X, whose equivalence classes correspond exactly to the sets of inputs that are *indistinguishable* based on some observed output in Y. Recall

that the kernel of a function $f : X \to Y$ is the *equivalence relation* κ_f over X, where $\forall x, x' \in X, (x, x') \in \kappa_f \iff f(x) = f(x')$.

An equivalence relation (ER) is a reflexive, symmetric and transitive binary relation over a set. We denote the set of all ERs over the set X by $ER(X)$. An ER partitions its domain to blocks referred to as its equivalence classes, where for any $x \in X$ and $R \in ER(X)$, the equivalence class of x is given by $[x]_R \triangleq \{x' \mid (x, x') \in R\}$. The partitioning of X by R is given by the set of equivalence classes of R defined as $[X]_R \triangleq \{[x]_R \mid x \in X\}$.

By considering their ability to distinguish elements of a set, ERs over that set can be arranged based on their information content. For any $R, R' \in ER(X)$, we say that R' contains more information than R, written $R \sqsubseteq R'$, iff for all $x, x' \in X, (x, x') \in R' \Rightarrow (x, x') \in R$. In other words, if x and x' are related by (belonging to the same equivalence class of) R', and are thus indistinguishable by R', then neither are they distinguishable by R because they are related by R. By the contrapositive, whatever pair R can distinguish (by not relating them), R' can also distinguish. The relation $\sqsubseteq$ is a *partial order* on ERs, and the partially-ordered set $\langle ER(X), \sqsubseteq \rangle$ is a complete lattice, whose greatest (most informative) element is id_X and the least element (least informative) is all_X, which are defined such that $\forall x, x' \in X, (x, x') \in id_X \iff x = x'$, and $\forall x, x' \in X, (x, x') \in all_X$. Since id_X only relates an element to itself, distinguishing every element in X from others, id_X contains the greatest information over X. But all_X contains no information since it relates *all* elements of X and cannot distinguish any pair of elements. The lattice *join* operation $\sqcup$ on ERs models information combination, and is defined for any $R, R' \in ER(X)$ such that $\forall x, x' \in X.(x, x') \in R \sqcup R' \iff (x, x') \in R \wedge (x, x') \in R'$. The join generalises to sets of ERs, so that for any $\mathcal{R} \subseteq ER(X)$ we have that, $\forall x, x' \in X.(x, x') \in \bigsqcup \mathcal{R} \iff \forall R \in \mathcal{R}.(x, x') \in R$. We shall define information erasure between a pair of ERs with respect to the greatest lower bound of the pair. The associated operation is the lattice *meet*, $\sqcap$, a dual of the information combination operator, $\sqcup$, that may be defined in terms of the join such that for any $R, R' \in ER(X)$ we have that $\forall x, x' \in X.(x, x') \in R \sqcap R' \iff (x, x') \in \bigsqcup \{R'' \in ER(X) \mid R'' \sqsubseteq R \wedge R'' \sqsubseteq R'\}$.

2.1 Qualitative Policies for Information Release and Erasure

We consider two information flow modes under the *what* dimension [22] of declassification: *release* ($\rightarrowtriangle$) and *erasure* ($\trianglelefteq$). For any two ERs $R, R' \in ER(X)$, we say that $R \rightarrowtriangle R'$ is an *information release* over the domain X whereby an agent with prior knowledge R is allowed to gain at most the information modelled by $R \sqcup R'$. Since $R \sqsubseteq R \sqcup R'$, $R \sqcup R'$ is an upper bound on the maximal information that the agent may gain, and thus $R \rightarrowtriangle R'$ constrains information release. If $R = R'$, then the release policy is analogous to *noninterference* at the information level R, since $R \rightarrowtriangle R$ prevents an agent from refining its prior knowledge R. Note that $R \rightarrowtriangle R$ is stronger than the standard notion of noninterference as the definition of noninterference does not take the agent's prior knowledge into account. However, $R \trianglelefteq R'$ is an *erasure policy*, whereby given some reference information R' contained in a body of data X, $R \sqcap R'$ is the maximal information that is allowed to be propagated. We may think of a system that conforms to the policy $R \trianglelefteq R'$ as an information eraser, which ensures that when it copies data from X, erases sufficient information that no more than $R \sqcap R'$ may be learnt from

the result. Thus, $R \sqcap R'$ is an upper bound on information that may be propagated to the destination, and since our reference R' is greater: $R' \sqsupseteq R \sqcap R'$, the propagation exhibits an information loss. In terms of an agent's knowledge, an agent that is observing a system that conforms to the erasure policy $R \twoheadleftarrow R'$ may not gain more than the information $R \sqcap R'$.

Definition 1 (Release and Erasure Policy Satisfaction). *Let $R, R' \in ER(X)$. We say that a system that is modelled by the function $f : X \to Y$, for some Y,* satisfies *(conforms to) the information flow policy $R \twoheadrightarrow R'$, written $f \vDash R \twoheadrightarrow R'$, if $\forall x, x' \in X, (x, x') \in R \sqcup R' \Rightarrow f(x) = f(x')$. Similarly, we say that the system* satisfies *the erasure policy $R \twoheadleftarrow R'$, written $f \vDash R \twoheadleftarrow R'$, if $\forall x, x' \in X, (x, x') \in R \sqcap R' \Rightarrow f(x) = f(x')$.*

We can describe the satisfaction of release and erasure policies by a system in terms of the kernel of a function f that models the system. Observe that from the definition of the kernel κ_f of f, we have $f(x) = f(x') \Rightarrow (x, x') \in \kappa_f$. By substituting this into Definition 1, we obtain $f \vDash R \twoheadrightarrow R' \Rightarrow \kappa_f \sqsubseteq R \sqcup R'$ and $f \vDash R \twoheadleftarrow R' \Rightarrow \kappa_f \sqsubseteq R \sqcap R'$.

2.2 Quantifying Information Erasure Policies

There is a natural underlying relationship between the partitioning of a set and quantitative information erasure (as well as release). In this section we show more directly this relationship and how it applies to erasure and release policies. We start again from a function $f : X \to Y$, which models a system of interest. Now let μ be a probability measure[2] over X. In conjunction with the probability measure μ, X may be regarded as a random variable that takes on various values according to the measure assigned by μ. Now, since Y is a function of X via f, f induces a probability measure over Y from μ. We abuse notations. Let $x \in X$ and $y \in Y$, we write $\mu(x)$ to represent the probability of selecting x as the input to the system, and write $\mu(y) = \sum_{x \in f^{-1}(y)} \mu(x)$ to be the probability of generating the output y. Similarly, $\mu(x|y)$ is the conditional probability that the input x was selected given the observation of output y, and $\mu(y|x)$ is the conditional probability that the output y will be produced, given the selection of input x. Since f is a function, we know that $\mu(y|x) = 1$ if $y = f(x)$, but $\mu(y|x) = 0$ otherwise.

Now given any $R \in ER(X)$, we say that R releases information about the domain X by partitioning it. Thus, given a probability measure μ over X, we define the quantitative information released from the space X, subject to its partitioning by R, to be $\mathcal{H}(\mu|R)$, defined as:

$$\mathcal{H}(\mu|R) \triangleq -\sum_{X' \in [X]_R} \mu(X') \log(\mu(X')) \tag{1}$$

The logarithm in (1) is to the base 2, and the definition of $\mathcal{H}(\mu|R)$ is a generalisation of the standard Shannon entropy [23], but defined over the equivalence classes of R. The standard Shannon entropy definition: $\mathcal{H}(\mu) = -\sum_{x \in X} \mu(x) \log(\mu(x)) = \mathcal{H}(\mu|id_X)$, is a special case of (1) where we condition by the identity equivalence relation id_X over X. It is easy to see that for any $R \in ER(X)$ and probability measure μ over X, $\mathcal{H}(\mu|R) \geq 0$.

[2] We may view μ as assigning probabilities to each element of X in accordance with its likelihood of being selected as an input.

Now, given $R, R' \in ER(X)$ and a probability measure μ over X, we can now define the information-theoretic quantification of the information release and erasure policies under the assumption μ as:

$$\begin{aligned} \mathcal{H}(\mu|R \twoheadrightarrow R') &\triangleq \mathcal{H}(\mu|R' \sqcup R) - \mathcal{H}(\mu|R) \\ \mathcal{H}(\mu|R \twoheadleftarrow R') &\triangleq \mathcal{H}(\mu|R') - \mathcal{H}(\mu|R \sqcap R') \end{aligned} \tag{2}$$

2.3 Illustrative Examples

Let us now illustrate by examples the use of the definitions of erasure and release presented above. Consider four systems which accept inputs from the domain $X = \{n \in \mathbb{Z} \mid -8 \leq n \leq 8, n \neq 0\}$ of integers, where the selection of inputs follows a uniform distribution, that is, $\forall x \in X, \mu(x) = \frac{1}{16}$. Suppose the first system, modelled by $f_1(x) = x$, echoes its input, releasing all the information; the second system, modelled by $f_2(x) = |x|$, erases the sign of its input; the third system releases the parity (or, erases everything but the parity), and is modelled by $f_3(x) = x \bmod 2$; and the fourth erases all information (releases nothing), and is modelled by the constant function $f_4(x) = 0$.

The kernel of f_1, is $\kappa_{f_1} = id_X$, the identity relation over X. Since the observer of the output of that system can completely determine the input to the system, the observer gains all the information about the input. This information release corresponds to the policy $all_X \twoheadrightarrow id_X$, which the system satisfies, and which allows the observer with no prior knowledge (all_X) to gain all information about inputs. Quantitatively, the information release that the policy $all_X \twoheadrightarrow id_X$ permits is calculated as $\mathcal{H}(\mu|all_X \twoheadrightarrow id_X) = 16(\frac{1}{16}\log(16)) - 1\log(1) = 4$. It is not surprising that the quantitative measure of the policy $all_X \twoheadrightarrow id_X$ is the total information contained in the input space X, and this fact holds for any assumption μ of the probability distribution of X. Since, the system modelled by f_1 releases all information, it cannot satisfy any non-trivial erasure policy. It only satisfies the trivial erasure policy $id_X \twoheadleftarrow id_X$, which erases no information because $\mathcal{H}(\mu|R \twoheadleftarrow R) = 0$ under any ER R and distribution μ. Which is just as well, because the system modelled by f_1 exhibits no erasure. The other extreme case is the system modelled by f_4, where $f_4(x) = 0$ for all input x. Thus, $\kappa_{f_4} = all_X$, and the relevant policy that the system satisfies is $all_X \twoheadrightarrow all_X$, which releases no information since $\mathcal{H}(\mu|all_X \twoheadrightarrow all_X) = 0$. However, the system exhibits total erasure over X since it satisfies the erasure policy $all_X \twoheadleftarrow id_X$, because $\mathcal{H}(\mu|all_X \twoheadleftarrow id_X) = 4$ is the total erasure of the information content of X.

Now the system modelled by f_2 erases only the *sign* of its input, which is a 1 bit information under the uniform distribution μ. The kernel of f_2 is κ_{f_2} defined such that $\forall x_1, x_2 \in X, (x_1, x_2) \in \kappa_{f_2} \iff |x_1| = |x_2|$. Hence the system modelled by f_2 satisfies both the information release policy $all_X \twoheadrightarrow \kappa_{f_2}$ and the erasure policy $\kappa_{f_2} \twoheadleftarrow id_X$. These are quantified under μ as $\mathcal{H}(\mu|all_X \twoheadrightarrow \kappa_{f_2}) = 3$ and $\mathcal{H}(\mu|\kappa_{f_2} \twoheadleftarrow id_X) = 1$ as we suspected: the system erases the 1 bit *sign* information, releasing the remaining 3 bits.

For f_3 we have the kernel κ_{f_3} defined such that $\forall x_1, x_2 \in X, (x_1, x_2) \in \kappa_{f_3} \iff x_1 \equiv x_2 (\bmod 2)$, releasing the 1 bit *parity* information and erasing all other information. As expected $\mathcal{H}(\mu|all_X \twoheadrightarrow \kappa_{f_3}) = 1$ and $\mathcal{H}(\mu|\kappa_{f_3} \twoheadleftarrow id_X) = 3$. Fig. 1 demonstrates the partitioning of the input domain by the various function kernels, and some of the corresponding erasure and release policies.

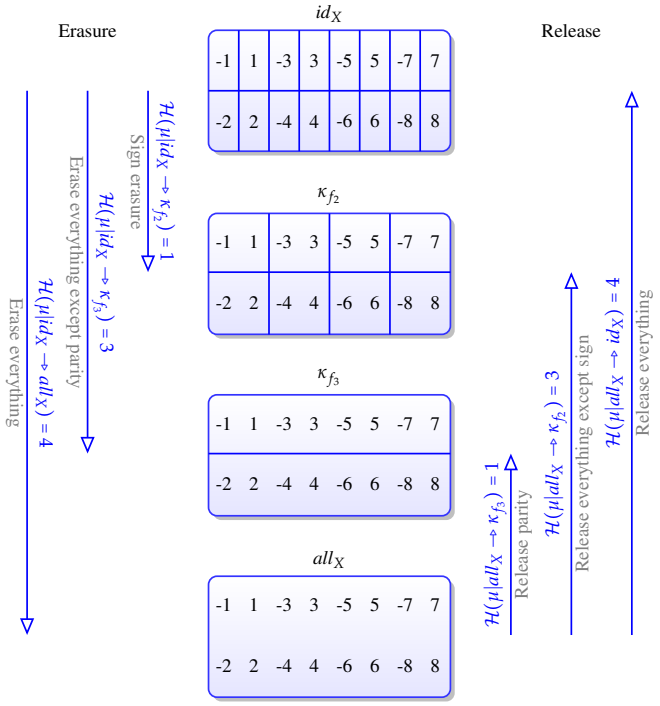

Fig. 1. Information Erasure and Release Policies (Uniform μ)

3 Some Properties of Quantified Information Erasure

The definitions in (2) of the quantification of release and erasure policies over a given space suggests that there is a duality between the erasure and release quantities with respect to the total information content over that space. More specifically the sum of the erasure and release quantities equals the total information content as Corollary 1 to Theorem 1 below shows. More generally, for any three ERs arranged on a chain, the sum of the quantified release between the bottom ER and the middle one and the erasure from the top ER to the middle is equal to the release (as well as the erasure) between the two extreme ERs as shown by Theorem 1.

Theorem 1. *For any chain of equivalence relations $R_1, R_2, R_3 \in ER(X)$ such that $R_1 \sqsubseteq R_2 \sqsubseteq R_3$ we have that $\mathcal{H}(\mu|R_1 \twoheadrightarrow R_2) + \mathcal{H}(\mu|R_2 \twoheadleftarrow R_3) = \mathcal{H}(\mu|R_1 \twoheadrightarrow R_3) = \mathcal{H}(\mu|R_1 \twoheadleftarrow R_3)$.*

A specialisation of Theorem 1 shows that for any $R \in ER(X)$, there is an associated maximum release and erasure policy with respect to R, which are $all_X \twoheadrightarrow R$ and $R \twoheadleftarrow id_X$ respectively. These complementary policies capture the release of at most information R about the domain X, when the observer is assumed to have no prior knowledge; or, dually, the deletion of all information in X, with the exception of R.

Corollary 1. *For any set X, equivalence relation R over X, and probability measure μ over X, we have that $\mathcal{H}(\mu|all_X \twoheadrightarrow R) + \mathcal{H}(\mu|R \twoheadleftarrow id_X) = \mathcal{H}(\mu)$.*

Now suppose that R is the kernel of a function $f : X \to Y$, which models a system of interest, then $all_X \twoheadrightarrow R$ is the information released by the system, and the quantification of this release is exactly the mutual information $\mathcal{I}(X;Y) = \mathcal{H}(X) - \mathcal{H}(X|Y)$ between random variables X and Y induced by some given probability measure over X. Thus, our definition of the quantitative measure of the release policy $all_X \twoheadrightarrow R$ coincides exactly with the standard measure of quantitative information release, establishing the utility of our definition. This relationship is shown in Theorem 2.

Theorem 2. *Let κ_f be the kernel of the function $f : X \to Y$, then $\mathcal{H}(\mu|all_X \twoheadrightarrow \kappa_f) = \mathcal{I}(X;Y)$. Furthermore, $\mathcal{H}(\mu|\kappa_f \twoheadleftarrow id_X) = \mathcal{I}(X) - \mathcal{I}(X;Y)$.*

Theorem 2 establishes a tight link between the partitioning of a function's domain by its kernel, the quantitative information flow, and the quantitative information erasure in any system modelled by that function. Note that the standard measure of quantitative information flow (see [5], for example) is the mutual information $\mathcal{I}(X;Y)$ between the domain and codomain of f. Furthermore, we observe that for a pair of comparable equivalence relations, erasure and release between them is identical.

Lemma 1. *Let $R, R' \in ER(X)$ such that $R \sqsubseteq R'$, and let μ be a probability measure over X. Then, $\mathcal{H}(\mu|R \twoheadrightarrow R') = \mathcal{H}(\mu|R \twoheadleftarrow R')$.*

Proof. Since $R \sqsubseteq R'$, it follows easily from the definition that $\mathcal{H}(\mu|R \twoheadleftarrow R') = \mathcal{H}(\mu|R') - \mathcal{H}(\mu|R \sqcap R') = \mathcal{H}(\mu|R') - \mathcal{H}(\mu|R) = \mathcal{H}(\mu|R \sqcup R') - \mathcal{H}(\mu|R) = \mathcal{H}(\mu|R \twoheadrightarrow R')$.

4 Some Limitations of Quantified Release and Erasure

A quantitative statement of security such as "*this system leaks at most c bits of information*" is quite appealing, but it has some practical limitations. For example, there is no way to characterise what sort of information the c bits refer to, leading to a problem referred to as information release *occlusion* [22], whereby the deliberate declassification of one piece information may mask another. To illustrate this problem, consider a simple example where input values are taken from a set $X = \{-4, -3, -2, -1, 1, 2, 3, 4\}$ where we desire to release no more than the *parity* information of the input, corresponding to the release policy $all_X \twoheadrightarrow R$, where $\forall x, x' \in X.(x, x') \in R \iff x \equiv x' \mod 2$. Now suppose the input probability distribution is such that $\mu(-2) = \mu(2) = \mu(-1) = \mu(1) = \frac{1}{4}$ and $\mu(-4) = \mu(4) = \mu(-3) = \mu(3) = 0$. We cannot capture the desired release requirement as a purely quantitative policy, because we must approximate by saying that the system may leak at most 1 bit of information. A system modelled by $f(x) = x \mod 2$ satisfies both the original qualitative requirement ($all_X \twoheadrightarrow R$) and the quantitative approximation of "at most 1 bit release", and is perhaps the sort of system that the policy writer had in mind. But then, there are other systems that release other information (possibly including the parity), that also qualify as not releasing more than 1 bit of information, for example $g(x) = |x|$. Clearly, g reveals both the parity of the

input, as well as its absolute value, and therefore does not satisfy the requirement to release at most the parity. However, the quantitative information released by g is 1 bit under μ. So, g satisfies the quantitative release requirement. This is a *release-occlusion* problem. We call the erasure analogue to this problem *erasure-occlusion*, whereby the erasure of one information masks another. These occlusion problems do not affect the ER policy model, which is rich enough to distinguish implementations such as those modelled by g from those modelled by f (since $g \not\models all_X \twoheadrightarrow R$, but $f \models all_X \twoheadrightarrow R$).

At the heart of the occlusion problem for the quantitative policy is the fact that the entropy definition is not sensitive to permutation of probabilities, because simply permuting the probabilities of the equivalence classes of a function's kernel does not affect the entropy (and thus information flow/erasure): the entropy definition is symmetric with respect to the permutation of labels. Furthermore, being numbers, the quantitative information measures do not have sufficient structure other than the total ordering of numbers to distinguish various release scenarios as the ER model can. So, quantitative policies alone may not be sufficient in practice. This limitation applies to all purely quantitative information flow models.

Problems also arise from the requirement to have (or assume) probability distributions in order to carry out quantitative analyses. One may simply not know the probability distribution of inputs, especially those about inputs from system attackers. Even when considering data under the control of the system owner, the *actual* probability distribution of the user-generated data is often unknown, and may have to be approximated or guessed. Such approximations and guessworks call into question the meaning of the quantitative results obtained during analysis, and whether the results paint the correct security picture. The belief-based approach of [6] may ameliorate this problem by considering assumed probability measures as beliefs about the actual system probabilities, and hence that the analysis result is correct modulo the difference between the actual system probability distribution and the analyst's assumption about it.

5 Related Work and Conclusion

The notion of partial information release has been extensively studied in language-based security [20] as a practical alternative to the traditional non-interference [10] policies. The problem is that noninterference is too restrictive for most practical purposes [19]. A common methodology for modelling partial information release is based on its quantification (typically via information theory) [1, 5, 15, 11, 24]. However as we noted earlier, quantitative approaches may suffer from the release occlusion problem. Qualitative approaches to modelling partial information release include [8, 14, 13, 21, 9]. The aforementioned quantitative and qualitative approaches all fall under the *what* dimension of declassification of the taxonomy proposed in [22], which considers four dimensions of information release and integrity; namely, *what* information is released, *where* the information is released, *who* releases information, and *when* information is released.

Qualitative models tend to be more precise in describing information flow and erasure as our approach demonstrates. In fact, as we have shown, given the relevant probability distributions, all the quantitative results in this paper are derivable from our quali-

tative definitions. However, we conjecture that the quantitative measure captures a probabilistic aspect of the input selection process in ways that the qualitative model cannot, and we suspect that this may serve as a useful policy point for specifying desirable probabilistic behaviour of the system and its environment. We are currently investigating this possibility. With respect to the interplay between qualitative and quantitative definitions of information security, one of our main results in this paper is a tighter link between the two. We have presented a model that captures, qualitatively and quantitatively, both partial release and partial erasure at the same time. We belief that combinations of both qualitative and quantitative models will begin to emerge, taking advantage of the strengths of each approach. In fact, new results [11, 15, 2] in quantitative information flow already use qualitative arguments such as "partitioning" and "families of disjoint sets" to develop the quantitative theory – alluding to some of the links that we show more directly in this paper. Indeed, the relationship between qualitative representation of information with equivalence relation (partition of a set) and information-theoretic quantitative representation has been studied recently in the context of information flow by [11, 15], but this is a relatively old mathematical idea [3, 16]. To our knowledge, we are the first to relate equivalence relations to erasure, quantifying this relationship.

Now, compared to information release in language-based settings, information erasure has only just begun to receive research attention [4, 12, 17, 25]. However, similarly to the research in the early days of language-based information-flow security that focussed on noninterference, the state of the art in information erasure is still at the level of *total erasure*, but there is already a recognition for the necessity of partial erasure in practice [25]. We tackle the more general problem of *partial erasure* in this paper, specifying qualitative and quantitative models for it. The authors of [12] observe that erasure should be studied in conjunction with noninterference in interactive systems, and [25] has begun to identify practical erasure patterns. This paper studies *what* information is erased, defining the end-to-end erasure achieved by a system. This is as opposed to, say, [4] or [12], which respectively study *when* and *where* information erasure takes place.

Quantitative notions of *contamination* and *suppression* are defined in [7] as facets of information integrity. The direct correspondence between the integrity notion of *contamination*, which measures the amount of untrusted input present in trusted output, and the *confidentiality* problem of determining the flow of information from sensitive input to public output as studied in quantitative information-flow, is well-known. In fact, the quantitative definition of contamination in [7] and the definition of leakage in [5] is the same, modulo labelling (untrusted vs. confidential, and trusted vs. public), suggesting that the difference is only cosmetic: specifically how the inputs and outputs are typed. Thus the difference is mainly the emphasis on the *who* dimension, i.e. whether they are trusted or not, or whether they are classified as "secret" or "public". Theorem 2 shows that the quantitative measure of contamination can be derived from our definition, which demonstrates its utility. Furthermore, the special case of Corollary 1 relating erasure and information release to the total information content derives the conclusion of Proposition 1 of [7] that the sum of leakage and suppression is constant. However, we suggest that the suppression measure as defined in [7] is the integrity analogue of information erasure with respect to the *what* dimension of information flow. Two sources of suppression are discussed in [7]: namely those that originate from a program's prob-

abilistic behaviour (that is, not necessarily untrusted) and those that originate directly from untrusted input (possibly from an attacker). We argue that this distinction is a *where*[3] versus *who*[4] dimension of information flow, which we do not study.

5.1 Conclusion

We have presented an extensional qualitative information security policy model that brings together partial information erasure and partial information release under a simple unified framework. We showed a tight link between the policy model and the standard information-theoretic definition of quantitative information flow. While the individual study of information release or erasure is not new, we believe that we present a compelling justification for their study under a single model. This is appealing because by incorporating partial information release and partial erasure under a single umbrella, the same analysis can be used to characterise both the information flow and erasure of a system in one go: we have shown that the *what* dimension of these concepts are duals. In other words, you buy one and get the other free. Specifically, given the functional model $f : X \to Y$ of a system (perhaps derived from a given attacker's interaction model), computing the information release and erasure is a matter of calculating the kernel κ_f of the function f; and these are given by $all_X \twoheadrightarrow \kappa_f$ and $\kappa_f \twoheadleftarrow id_X$ respectively. The corresponding measures $\mathcal{H}(\mu|all_X \twoheadrightarrow \kappa_f)$ and $\mathcal{H}(\mu|\kappa_f \twoheadleftarrow id_X)$ then allow us to quantify information release and erasure under an input distribution μ.

Our definitions consider a foundational issue which makes clearer the extensional relationship between partial erasure and partial release. These definitions will enable the development of policies for secure information erasure and release, with potential applications to existing and new problem areas. We have however not defined verification or enforcement mechanisms for information erasure and release in systems. These are areas of future work. In particular, an avenue for further research is the design of frameworks to derive relevant system models from system specifications such as process calculi, or from software source and binary code; and the static analysis of the same for conformance to erasure policies.

References

1. M. Backes. Quantifying probabilistic information flow in computational reactive systems. In *Proceedings of 10th European Symposium on Research in Computer Security*, volume 3679 of *Lecture Notes in Computer Science*, pages 336–354. Springer, September 2005.
2. M. Backes, M. Berg, and B. Köpf. Non-uniform distributions in quantitative information-flow. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '11, pages 367–375, New York, NY, USA, 2011. ACM.
3. P. Billingsley. *Ergodic Theory and Information*. J. Wiley and Sons, New York, 1965.
4. S. Chong and A.C. Myers. Language-based information erasure. In *Computer Security Foundations, 2005. CSFW-18 2005. 18th IEEE Workshop*, pages 241 – 254, june 2005.
5. D. Clark, S. Hunt, and P. Malacaria. A static analysis for quantifying information flow in a simple imperative language. *Journal of Computer Security*, 15(3):321–371, 2007.

[3] Because the suppression originates from *within* the program.

[4] Because the *untrusted* input influences suppression.

6. M. R. Clarkson, A. C. Myers, and F. B. Schneider. Quantifying information flow with beliefs. *Journal of Computer Security*, 17(5):655–701, 2009.
7. M. R. Clarkson and F. B. Schneider. Quantification of integrity. In *Proceedings of the 23rd IEEE Computer Security Foundations Symposium, CSF 2010, Edinburgh, United Kingdom*, pages 28–43. IEEE Computer Society, 17-19 July 2010.
8. E. S. Cohen. Information transmission in sequential programs. In Richard A. DeMillo, David P. Dobkin, Anita K. Jones, and Richard J. Lipton, editors, *Foundations of Secure Computation*, pages 297–335. Academic Press, 1978.
9. R. Giacobazzi and I. Mastroeni. Abstract non-interference: parameterizing non-interference by abstract interpretation. In *Proceedings of the 31st ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 186–197. ACM Press, 2004.
10. J. A. Goguen and J. Meseguer. Security policies and security models. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 11–20, Oakland, CA, April 1982. IEEE Computer Society Press.
11. J. Heusser and P. Malacaria. Applied quantitative information flow and statistical databases. In Pierpaolo Degano and Joshua Guttman, editors, *Formal Aspects in Security and Trust*, volume 5983 of *Lecture Notes in Computer Science*, pages 96–110. Springer, 2010.
12. S. Hunt and D. Sands. Just forget it: The semantics and enforcement of information erasure. In *Proc. 17th European Symposium on Programming (ESOP'08)*, volume 4960 of *Lecture Notes in Computer Science*, Budapest, Hungary, March 2008. Springer-Verlag.
13. R. Joshi and K. R. M. Leino. A semantic approach to secure information flow. *Science of Computer Programming*, 37(1-3):113–138, 2000.
14. J. Landauer and T. Redmond. A lattice of information. In *Proceedings of the Computer Security Foundations Workshop VI (CSFW '93)*, pages 65–70, Washington - Brussels - Tokyo, June 1993. IEEE.
15. P. Malacaria. Risk assessment of security threats for looping constructs. *Journal of Computer Security*, 18(2):191–228, 2010.
16. Y. Nakamura. Entropy and semivaluations on semilattices. *Kōdai Mathematical Seminar Reports*, 22(4):443–468, 1970.
17. A. Nanevski, A. Banerjee, and D. Garg. Verification of information flow and access control policies with dependent types. In *IEEE Symposium on Security and Privacy*, pages 165–179. IEEE Computer Society, 2011.
18. PCI Security Standards Council LLC. Payment Card Industry (PCI) Data Security Standard: Requirements and Security Assessment Procedures, October 2010.
19. P. Ryan, J. McLean, J. Millen, and V. Gligor. Non-interference, who needs it? In *14th IEEE Computer Security Foundations Workshop (CSFW '01)*, pages 237–240, Washington - Brussels - Tokyo, June 2001. IEEE.
20. A. Sabelfeld and A. C. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications*, 21(1):5–19, January 2003.
21. A. Sabelfeld and D. Sands. A Per Model of Secure Information Flow in Sequential Programs. *Higher-Order and Symbolic Computation*, 14(1):59–91, March 2001.
22. A. Sabelfeld and D. Sands. Declassification: Dimensions and principles. *Journal of Computer Security*, 2007.
23. C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.
24. G. Smith. Quantifying information flow using min-entropy. In *Quantitative Evaluation of Systems (QEST), 2011 Eighth International Conference on*, pages 159 –167, sept. 2011.
25. F. D. Tedesco, S. Hunt, and D. Sands. A semantic hierarchy for erasure policies. In Sushil Jajodia and Chandan Mazumdar, editors, *Information Systems Security - 7th International Conference, ICISS 2011, Kolkata, India, December 15-19, 2011, Procedings*, volume 7093 of *Lecture Notes in Computer Science*, pages 352–369. Springer, 2011.