A Better Composition Operator for Quantitative Information Flow Analyses

Kai Engelhardt(✉)

CSE, UNSW, Sydney, Australia
kaie@cse.unsw.edu.au

Abstract. Given a description of the quantitative information flow (qif) for components, how can we determine the qif of a system composed from components? We explore this fundamental question mathematically and provide an answer based on a new composition operator. We investigate its properties and prove that it generalises existing composition operators. We illustrate the results with a fresh look on Chaum's dining cryptographers. We show that the new operator enjoys various convenient algebraic properties and that it is well-behaved under composition refinement.

1 Introduction

In the area of *quantitative information flow (qif)* analysis, we concern ourselves with measuring or deriving the amount of information leaking from systems. A popular model of systems in qif is that of channel matrices which contain precise descriptions of the probabilities of observing certain public outputs given certain secret inputs.

We refer to the survey by Smith [27] for further motivation of this general direction in qif research. Compared to the literature, we use a slightly different definition of channels to prepare for the various composition operators later. Our change is similar to a move from opaque states as they are common in automata theory on the one hand to program states as mappings from variable names to values as they are common in treatments of program semantics on the other hand.

In Sect. 2 we define our model including the new operator $\bowtie$ and argue that it is a reasonable choice for a composition operator. We do so by showing firstly that $\bowtie$ offers a new and arguably elegant decomposition of the well-known dining cryptographers example. This decomposition uses simple laws from a channel algebra for equality between channels. In Sect. 3 a more interesting algebra emerges when replacing equality by composition refinement, a leakage-reducing notion of refinement on channels. We prove that $\bowtie$ again enjoys interesting properties. We show in Sect. 4 that $\bowtie$ subsumes various existing composition operators and that its algebraic laws specialise to laws for the existing operators.

2 Mix Composition

Notation. We write $\mathbb{B} = \{\mathtt{0}, \mathtt{1}\}$ for the Booleans. By $[0, 1]$ we denote the closed real interval between 0 and 1. For $a, b \in \mathbb{N}$ we define $a..b = \{\, x \in \mathbb{N} \mid a \leq x \leq b \,\}$.

S.N. Foley et al. (Eds.): ESORICS 2017, Part I, LNCS 10492, pp. 446–463, 2017.
DOI: 10.1007/978-3-319-66402-6_26

We write $f \downarrow_S$ for the *domain restriction* $\lambda s : S.f(s)$ of function f by set S. Our channels map named inputs to named outputs. These names correspond to wires in circuits and variables in programs. Each name is associated with a domain of possible values. To compose channels we require the names of their wires/variables so we know which of their inputs and outputs hook up. Formally, if S_k is a set for each $k \in K$, we write $\bigotimes_{k \in K} S_k$ for the set of functions $f : K \longrightarrow \bigcup_{k \in K} S_k$ satisfying $f(k) \in S_k$ for all $k \in K$. All our logarithms are base 2. Binary operators that are commutative and associative such as our forthcoming composition operator are implicitly lifted to indexed families of arguments, just as $+$ is lifted to $\sum$, only that we don't use a separate symbol.

Channels. Not surprisingly, functions in $\bigotimes_{k \in K} S_k$ resemble states in program semantics. Programs or system components transform states to states according to their function. In qif research, programs and systems are commonly called channels and they map (secret) input states to distributions of (observable) output states.

We assume that secret inputs have some prior distribution which is known to observers. A channel can then be understood as mapping each prior to a posterior distribution on the outputs, which in turn can be understood as a distribution of distributions of inputs. We also assume that the channel itself is known to observers. We define channels formally.

Definition 1 (Channel). *Let $\mathcal{V}$ be a set we call* variables*. Let $\mathcal{X} = (X_w)_{w \in \mathcal{V}}$ be a family of nonvoid finite sets, the* domains *of variables. Given a set V of variables, we denote their joint domain $\bigotimes_{v \in V} X_v$ by $d(V)$.*

A $(\mathcal{V}, \mathcal{X})$-channel *$(I, O, c)$* (from inputs named I to outputs named O) *consists of a finite set $I \subseteq \mathcal{V}$ of* input variables*, a finite set $O \subseteq \mathcal{V}$ of* output variables*, and a* channel matrix *$c \in [0, 1]^{d(I) \times d(O)}$ such that each row adds up to one, that is: $\forall x \in d(I) \left(\sum_{y \in d(O)} c_{x,y} = 1 \right)$.*

Denote the set of $(\mathcal{V}, \mathcal{X})$-channels from inputs named I to outputs named O by $\mathcal{C}_{\mathcal{V},\mathcal{X}}(I, O)$. A channel is called deterministic *when its matrix contains only zeros and ones.*

Note that I and O need not be disjoint. We often identify channels with their channel matrices, assuming that the input and output names are understood. Next we define a small set of basic channels that will be useful in later examples and algebraic laws. Write $\mathbb{O}_{I,O}$ for the *unit channel* in $\mathcal{C}_{\mathcal{V},\mathcal{X}}(I, O)$ that maps inputs named I to outputs named O in a uniform manner, i.e., $(\mathbb{O}_{I,O})_{x,y} = \frac{1}{|\mathrm{d}(O)|}$ for all $x \in \mathrm{d}(I)$ and $y \in \mathrm{d}(O)$. A special case are the unit channels where $O = \emptyset$. They have no designated output variables. Hence their channel matrices are column vectors full of ones. These are the only unit channels that are deterministic. Let $\mathbb{I}_V$ denote the *identity channel* in $\mathcal{C}_{\mathcal{V},\mathcal{X}}(V, V)$ with the matrix given by $(\mathbb{I}_V)_{x,y} = \delta_{x,y}$ where δ is the Kronecker delta. Identity channels are deterministic. Renaming channels are a generalisation of identity channels. Firstly, as the name suggests, renaming channels can rename the variables. Secondly, they allow a widening of the output variables' domains. More formally, if $I, O \subseteq \mathcal{V}$

and $f : \mathrm{d}(I) \longrightarrow \mathrm{d}(O)$ is injective, we define the *renaming channel (from I to O using f)* $\mathrm{R}^f_{I,O} \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,O)$ by $(\mathrm{R}^f_{I,O})_{x,y} = \delta_{f(x),y}$. We omit the injection if it is the identity function. We write $\mathrm{R}^f_{i,o}$ for $\mathrm{R}^f_{\{i\},\{o\}}$. We write injections f as expressions in the variables.

Example 2. Let $\mathcal{V} = \{i, o\}$ and $X_i = X_o = \mathbb{B}$. A 1-bit copying channel from i to o would be written as $\mathrm{R}_{i,o}$. Its channel matrix is the identity matrix $\left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)$. Next consider a channel $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(\{i\},\{o\})$ given by the matrix $\left(\begin{smallmatrix}1/3&2/3\\0&1\end{smallmatrix}\right)$. For instance, the probability of observing output $o = \mathtt{1}$ of channel A when the secret input is $i = \mathtt{0}$ is $A_{\mathtt{0},\mathtt{1}} = 2/3$.

Consider the distribution $\pi = (1/4, 3/4)$ on the Booleans. Multiplying prior π as a row vector with A's channel matrix yields the posterior distribution $\pi A = (1/12, 11/12)$ which means that with π as prior we expect to observe the output $o = \mathtt{1}$ with probability $11/12$. Multiplying each cell of A's matrix with the prior probability of its row according to π yields the *joint matrix* $\left(\begin{smallmatrix}1/12&2/12\\0&3/4\end{smallmatrix}\right)$, i.e., a distribution on input/output pairs. Normalising the columns results in $\left(\begin{smallmatrix}1&2/11\\0&9/11\end{smallmatrix}\right)$. Its column labelled $y = \{o \mapsto b\}$ for $b \in \mathbb{B}$ can now be read as a distribution on the secret input, given the output is y. For instance, if $y(o) = \mathtt{1}$, the input must have been $\{i \mapsto \mathtt{0}\}$ with probability $2/11$.

Next we define our new composition operator.

Definition 3 (Mix-composition). *Let $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,O)$ and $B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(J,P)$. We call them* $\bowtie$-compatible *if, for all $x \in d(I \cup J)$ there exists a $y \in d(O \cup P)$ such that both $A_{x\downarrow_I, y\downarrow_O}$ and $B_{x\downarrow_J, y\downarrow_P}$ are positive. If A and B are $\bowtie$-compatible we define their* mix-composition *as the channel $A \bowtie B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I \cup J, O \cup P)$ by*

$$(A \bowtie B)_{x,y} = \frac{A_{x\downarrow_I, y\downarrow_O} B_{x\downarrow_J, y\downarrow_P}}{\sum_{z \in d(O \cup P)} A_{x\downarrow_I, z\downarrow_O} B_{x\downarrow_J, z\downarrow_P}} \enspace ,$$

for all $x \in d(I \cup J)$ and $y \in d(O \cup P)$.

Note that our mix composition unifies

- inputs of the same name to model components sharing input variables and
- outputs with the same name to model that two components *collude* on such outputs. The components implicitly rule out contradicting observations with $\bowtie$-compatibility ensuring that there is at least one consistent observation per secret input.

In the remainder we typically assume $\bowtie$-compatibility for our results.

Example 4. Let $X_i = X_o = \mathbb{B}$. Consider the two 1-bit channels $A = \mathrm{R}_{i,o}$ and $B = \mathrm{R}^{(o=\neg i)}_{i,o} \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(\{i\},\{o\})$. (The expression $(o = \neg i)$ is shorthand for the injection $\lambda b : \mathrm{d}(\{i\}).\{o \mapsto \neg b(i)\}$.) Their channel matrices are $\left(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right)$, respectively. But their attempted $\bowtie$ composition matrix $\left(\begin{smallmatrix}A_{\mathtt{0},\mathtt{0}}B_{\mathtt{0},\mathtt{0}}&A_{\mathtt{0},\mathtt{1}}B_{\mathtt{0},\mathtt{1}}\\A_{\mathtt{1},\mathtt{0}}B_{\mathtt{1},\mathtt{0}}&A_{\mathtt{1},\mathtt{1}}B_{\mathtt{1},\mathtt{1}}\end{smallmatrix}\right) = \left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right)$ indicates that they are not $\bowtie$-compatible. Intuitively A and B attempt to collude on outputs but fail to agree.

We collect some sanity checks in[1] our

Proposition 5. *When channels are $\bowtie$-compatible*

1. *mix composition is well-defined, commutative, and associative;*
2. *mix composition of deterministic channels is again deterministic;*
3. *mix composition is idempotent when restricted to deterministic channels.*

Example 6. To see that mix composition is not necessarily idempotent on arbitrary channels, recall channel A from Example 2. We compute the channel matrix of $A \bowtie A$ as $\left(\begin{smallmatrix} 1/5 & 4/5 \\ 0 & 1 \end{smallmatrix}\right)$ the top row of which is clearly different from A's. The same example demonstrates that in general row normalisation is required. Without it, the "channel" matrix of $A \bowtie A$ had been $\left(\begin{smallmatrix} 1/9 & 4/9 \\ 0 & 1 \end{smallmatrix}\right)$ with row sum $5/9$ for the top row.

An exact version of Proposition 5.3 is

Proposition 7. *Let $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,O)$. Mix composition is idempotent on A iff each row of A has a unique non-zero value:*

$$A \bowtie A = A \quad \Leftrightarrow \quad \forall x \in d(I)\,(\exists v \in (0,1]\,(\forall y \in d(O)\,(A_{x,y} \in \{0,v\}))) \enspace .$$

Iterated self-composition of channels has limits that are non-trivial when the condition for idempotence is not met. Roughly speaking, self-composition is a form of *amplification* resembling established results in complexity theory such as the amplification lemma for **BPP**. In the limit, only the maximal values in each row survive—everything else becomes zero.

Proposition 8. *Let $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,O)$. Define $A^{(k)} = \bowtie_{i=1}^{k} A$ for all $k \in \mathbb{N}$. The limit $\lim_{k\to\infty} A^{(k)}$ exists and is given by the channel matrix with cells*

$$A^{(\infty)}_{x,y} = \begin{cases} \frac{1}{|\{\, y' \in d(O) \mid A_{x,y'} = \max_{y'' \in d(O)} A_{x,y''} \,\}|} & \text{if } A_{x,y} = \max_{y' \in d(O)} A_{x,y'} \\ 0 & \text{otherwise.} \end{cases}$$

In many practical cases, row normalisation is not required when computing mix compositions.

Proposition 9. *If A and B are deterministic and $\bowtie$-compatible, or if their output names are disjoint, then row normalisation is not required, that is, $(A \bowtie B)_{x,y} = A_{x\downarrow_I, y\downarrow_O} \cdot B_{x\downarrow_J, y\downarrow_P}$, for all $x \in d(I \cup J)$ and $y \in d(O \cup P)$.*

A simple distributivity result holds whenever a particular channel in the composition is deterministic.

Proposition 10. *Let $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,O)$ be deterministic. Let $B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(J,P)$ and $C \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(K,Q)$. Then $A \bowtie (B \bowtie C) = (A \bowtie B) \bowtie (A \bowtie C)$.*

[1] Proofs are given in the Appendix.

Example 11. To see that determinism of A is required in general for the distributivity result to hold, recall once again channel A from Example 2. In Example 6 we showed that $A \neq A \bowtie A$. Next we note that $A \bowtie \mathbb{O}_{\{i\},\emptyset} = A$ and that $\mathbb{O}_{\{i\},\emptyset} \bowtie \mathbb{O}_{\{i\},\emptyset} = \mathbb{O}_{\{i\},\emptyset}$. Clearly, $A \bowtie (\mathbb{O}_{\{i\},\emptyset} \bowtie \mathbb{O}_{\{i\},\emptyset}) = A \neq A \bowtie A = (A \bowtie \mathbb{O}_{\{i\},\emptyset}) \bowtie (A \bowtie \mathbb{O}_{\{i\},\emptyset})$.

Proposition 12. $\mathbb{I}_I \bowtie \mathbb{I}_J = \mathbb{I}_{I\cup J}$

The other fundamental channel composition operator is sequential, or cascading, composition.

Definition 13. *For $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I, M)$ with channel matrix c and $B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(M, O)$ with channel matrix d we define their* sequential composition $A; B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I, O)$ *by the channel matrix cd.*

2.1 Example: Dining Cryptographers

Chaum [7] introduced the dining cryptographers problem and offered a protocol as solution which has been studied to the extent that adding to the existing body of analyses induces a considerable amount of guilt. Here we investigate a slight variation of the problem insofar as we study the effect of collusion among the n cryptographers.

Let us write $\otimes$ for exclusive-or, $\oplus$ and $\ominus$ for addition, resp., subtraction modulo n.

A gaggle of n cryptographers named $0..n-1$ sit around a dinner table in clockwise order. When it's time to pay, the waiter informs them that the bill has already been paid. Either exactly one of the cryptographers paid for the dinner or the NSA did. The problem is to figure out whether the NSA paid or not, without compromising the anonymity of the paying cryptographer if the NSA didn't.

Chaum's protocol solves the problem as follows. Each cryptographer m secretly flips a coin. The outcome c_m is then shared only with the cryptographer $m \oplus 1$ immediately to their left. Each cryptographer m then announces the exclusive-or of three Boolean values: the two known coin values, c_m and $c_{m\ominus 1}$, and whether m paid. The exclusive-or of all announcements is true if one of the cryptographers paid and false if the NSA paid.

We begin by describing some of the variables and their domains. The coins named $c_0, \ldots, c_{n-1} \in \mathcal{V}$ have Boolean domains, that is, $X_{c_m} = \mathbb{B}$ for $m \in 0..n-1$. Who paid, named $p \in \mathcal{V}$, ranges over $X_p = 0..n$, where the value n denotes that the NSA paid. The announcements, named $a_0, \ldots, a_{n-1} \in \mathcal{V}$ also have Boolean domains. We model each cryptographer m as a channel $C^{(m)} \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(\{p, c_{m\ominus 1}, c_m\}, \{a_m\})$ with the channel matrix given by

$$C^{(m)}_{x,y} = \delta_{x(c_{m\ominus 1})\otimes x(c_m)\otimes(x(p)=m),y(a_m)} \ .$$

This matrix has $2^2(n+1)$ rows and two columns. We note that $C^{(m)}$ is deterministic. The view of an outside observer is

$$\mathrm{DC}_n = \mathop{\bowtie}_{m=0}^{n-1} C^{(m)} \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(\{p, c_0, \ldots, c_{n-1}\}, \{a_0, \ldots, a_{n-1}\}) \ .$$

(See Fig. 1.) Its channel matrix has $2^n(n+1)$ rows and 2^n columns and, as a mix composition of deterministic channels, is deterministic.

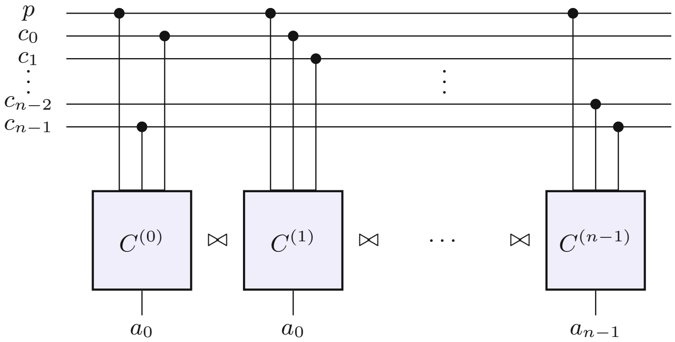

Fig. 1. Dining cryptographers as mix composition.

Cryptographer i observes not only DC_n but also the two coins c_i and $c_{i\ominus 1}$. In other words, cryptographer i's view of the situation is $C_i = \mathrm{DC}_n \bowtie \mathbb{I}_{\{c_i, c_{i\ominus 1}\}}$. (Technically, i also observes whether $p = i$ but that's already captured by the exclusive-or of its own three outputs, a_i, c_i, and $c_{i\ominus 1}$. An output that is a function of other outputs can be safely omitted.)

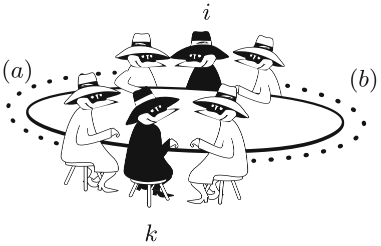

Fig. 2. Two colluding cryptographers i and k can eliminate one contiguous section, (a) or (b), as potential payers.

When considering *two* colluding cryptographers who pool their knowledge, we expect them to be able to divide the remaining cryptographers into two groups: (a) those to the right of i and to the left of k and (b) those to the left of i and to the right of k. (See Fig. 2.) The interesting result is that, in case one of the remaining cryptographers paid, the colluding cryptographers acquire (distributed) knowledge to which of the groups, (a) or (b), the payer belongs, thereby eliminating all members of the other group from the possible payers.

If one of the two groups is empty then it cannot contain the payer, meaning that i and k learn less.

As a channel, i and k together have the view $C_i \bowtie C_k$. Note that if i and k are adjacent (and $n > 2$) then they observe *three* coins—otherwise they observe *four* coins. Intuitively, this already implies that the information leaked in the former situation is less than that leaked in the latter. Using Proposition 5 we simplify as follows.

$$\begin{aligned} C_i \bowtie C_k &= \mathrm{DC}_n \bowtie \mathbb{I}_{\{c_i, c_{i\ominus 1}\}} \bowtie \mathrm{DC}_n \bowtie \mathbb{I}_{\{c_k, c_{k\ominus 1}\}} \\ &= \mathrm{DC}_n \bowtie \mathbb{I}_{\{c_i, c_{i\ominus 1}\}} \bowtie \mathbb{I}_{\{c_k, c_{k\ominus 1}\}} \end{aligned}$$

which, with Proposition 12, simplifies to

$$= \mathrm{DC}_n \bowtie \mathbb{I}_{\{c_i, c_{i\ominus 1}, c_k, c_{k\ominus 1}\}} \quad .$$

3 Channel Refinement with Mix Composition

We briefly recall the relevant definitions of leakage-related notions. Details and pointers to their origin can again be found e.g. in [27]. The (multiplicative) *min-capacity* of a channel $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I, O)$, denoted $\mathcal{ML}(A)$, is the maximum min-entropy leakage of A over all priors π: $\sup_\pi \log(\frac{V[\pi,A]}{V[\pi]})$. As proved by Braun et al. [6], the min-capacity of A can be computed as the logarithm of the sum of the column maximums of A, and it is always realised on a uniform prior π, so we have $\mathcal{ML}(A) = \log \sum_{y \in \mathrm{d}(O)} \max_{x \in \mathrm{d}(I)} A_{x,y}$.

Example 14 (Dining Cryptographers cont'd). Returning to the example in Sect. 2.1, we compute the min-capacities of various channels in case the number of cryptographers is $n = 4$.

Each individual cryptographer's channel has the same $\mathcal{ML}(C^{(m)}) \simeq 1.0$ because the channel is deterministic and has two columns. As a deterministic channel with 2^4 non-zero columns, the channel DC_4 has the min-capacity 4.0. Once we add, say, cryptographer 1's observation we obtain $\mathcal{ML}(\mathrm{DC}_4 \bowtie \mathbb{I}_{\{c_0,c_1\}}) \simeq 5.58$. Adding a second adjacent cryptographer's observation (as on the left of Fig. 3), say cryptographer 2's, the min-capacity goes up to $\mathcal{ML}(\mathrm{DC}_4 \bowtie \mathbb{I}_{\{c_0,c_1,c_2\}}) = 6.0$ whereas with a second cryptographer sitting opposite (as on the right of Fig. 3) $\mathcal{ML}(\mathrm{DC}_4 \bowtie \mathbb{I}_{\{c_0,c_1,c_2,c_3\}})$ goes up to approx. 6.32.

A more general notion of the leakage of channels is that of g-leakage [2]. We recall the relevant definitions here, adapted to our channels.

Definition 15. *Given a non-void set $\mathcal{W}$ of guesses and a finite set of inputs I, a* gain function *is a function $g : \mathcal{W} \times d(I) \longrightarrow [0, 1]$. The value $g(w, x)$ represents the gain of the attacker when the secret value is x and he makes a guess w on x. Given a gain function g and a prior π on $d(I)$, the* prior g-vulnerability *is $V_g(\pi) = \max_{w \in \mathcal{W}} \sum_{x \in d(I)} \pi(x) g(w, x)$. Given $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I, O)$, the* posterior g-vulnerability *is $V_g(\pi, A) = \sum_{y \in d(O)} \max_{w \in \mathcal{W}} \sum_{x \in d(I)} \pi(x) A_{x,y} g(w, x)$. The prior and posterior* g-entropy *is $H_g(\pi) = -\log V_g(\pi)$, resp., $H_g(\pi, A) = -\log V_g(\pi, A)$. The* g-leakage *is their difference $\mathcal{L}_g(\pi, A) = H_g(\pi) - H_g(\pi, A)$.*

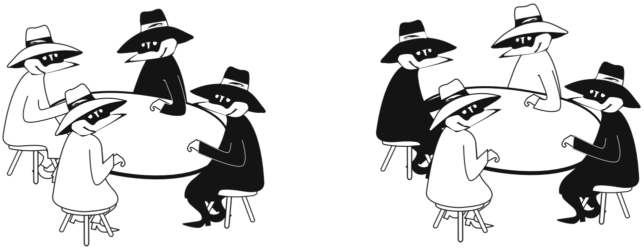

Fig. 3. Different seating arrangements of otherwise equal cryptographers result in different leakage from the collusion.

Example 16 (Dining Cryptographers cont'd). Continuing on from Example 14, we compute the g-leakage of various channels. An adversary curious about who paid observes just cryptographer m. We assume a uniform prior, guesses $\mathcal{W} = 0..n$, and a gain function given by $g(w, x) = \delta_{w,x(p)}$: the adversary gains 1 iff she guesses the payer exactly. That observing just one cryptographer is futile is indicated by $\mathcal{L}_g(\pi, C^{(m)}) = 0$. This remains unchanged if the model is modified such that the adversary only guesses whether the NSA paid or not, using $\mathcal{W} = \mathbb{B}$ and $g_{\mathbb{B}}(w, x) = \delta_{w,x(p)=n}$. With that goal the adversary is better off observing all n cryptographers. Assuming again a uniform prior we obtain $V_{g_{\mathbb{B}}}(\pi) = {}^{n}/_{n+1}$ and $V_{g_{\mathbb{B}}}(\pi, \mathrm{DC}_n) = 1$ which results in $\mathcal{L}_{g_{\mathbb{B}}}(\pi, \mathrm{DC}_n) = \log({}^{n+1}/_{n})$. Returning to the task of guessing who paid, but removing the gain in case it was the NSA, we consider $\mathcal{W} = 0..n-1$ and calculate again that this is futile: $\mathcal{L}_g(\pi, \mathrm{DC}_n) = 0$. This remains unchanged when we also remove the gain for cryptographer m and study what leaks to m about who paid (other than him and the NSA): with $\mathcal{W} = 0..n-1 \setminus \{m\}$ we have $\mathcal{L}_g(\pi, C_m) = 0$. Even if two adjacently seated cryptographers collude (as on the left of Fig. 3), we still have $\mathcal{L}_g(\pi, C_m \bowtie C_{m\oplus 1}) = 0$ if $n > 3$ and both are removed from the guesses. If, however, they are separated on both sides by at least one cryptographer (as on the right of Fig. 3) then we find that $\mathcal{L}_g(\pi, C_m \bowtie C_{m\oplus 2}) > 0$.

This completes the illustration of the fact that there's no obvious way to calculate relevant vulnerability measures of $\bowtie$-composed systems from the vulnerabilities of their components. We follow McIver et al. [21] in defining a robust leakage order on channels with the same inputs. The order is based on another familiar composition operator, sequential composition.

Definition 17. *Let $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I, O)$ and $B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I, M)$. We say that A* refines *B (written $B \sqsubseteq A$) if there exists a (post-processing) channel $C \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(M, O)$ such that $A = B; C$. We write $A \equiv B$ whenever A and B refine each other.*

As shown by MvIver et al. [21][2], $A \sqsubseteq B$ iff the g-leakage of A is never smaller than that of B, for any prior π and gain function g.

We list some immediate consequences of these definitions.

[2] and then discovered by Geoffrey Smith to be already contained in [5].

Proposition 18. *Unit channels are the top elements in the refinement order and the neutral elements of mix composition. Identity channels are the bottom elements in the refinement order and weak zeros of mix composition. More formally, let $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,O)$. Let $Q \subseteq \mathcal{V}$ be finite. Let $J \subseteq I$.*

$$\mathbb{I}_I \sqsubseteq A \tag{1}$$

$$A \sqsubseteq \mathbb{O}_{I,Q} \tag{2}$$

$$\mathbb{I}_I \equiv \mathbb{I}_I \bowtie A \tag{3}$$

$$\mathbb{O}_{J,Q} \bowtie A \equiv A \tag{4}$$

More interestingly, we have that $\bowtie$ is monotone w.r.t. composition refinement if no outputs are fused.

Theorem 19. *If $A \sqsubseteq A'$ and $B \sqsubseteq B'$ and neither A and B nor A' and B' share output names, then $A \bowtie B \sqsubseteq A' \bowtie B'$.*

Example 20. To see that $\bowtie$ is in general not $\sqsubseteq$-monotone when output names are shared, recall channel A from Example 2. Let $B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(\{i\},\{p\}) = A;\mathrm{R}_{o,p}$. Clearly $A \equiv B$. Let us compare $A \bowtie A$ to $A \bowtie B = \left(\begin{smallmatrix} 1/9 & 2/9 & 2/9 & 4/9 \\ 0 & 0 & 0 & 1 \end{smallmatrix}\right)$. Both $\bowtie$ compositions are defined, that is, A is compatible with itself and B. Solving $(A \bowtie A);X = A \bowtie B$ for X yields the unique solution $X = \left(\begin{smallmatrix} 1/3 & 2/3 & 2/3 & -2/3 \\ 0 & 0 & 0 & 1 \end{smallmatrix}\right)$, which is not a channel matrix because $-2/3 \notin [0,1]$. Hence $A \bowtie A \not\sqsubseteq A \bowtie B$.

The equation $(A \bowtie B);X = A \bowtie A$ is solved by $X = \begin{pmatrix} 9/25 & 16/25 \\ 9/25 & 16/25 \\ 9/25 & 16/25 \\ 0 & 1 \end{pmatrix}$, which is a channel matrix, hence $A \bowtie B \sqsubseteq A \bowtie A$.

Combining Theorem 19 with Proposition 18. (2) yields

Corollary 21. *Let $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,O)$ and $B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(J,P)$. Then*

$$A \bowtie B \sqsubseteq A \bowtie \mathbb{O}_{J\setminus I,\emptyset} \ ,$$

provided $O \cap P = \emptyset$.

Refining a channel to a mix composition means that the former refines to each of the components of the latter when a little care is taken with extra inputs.

Theorem 22. *Let $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,O)$, $B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(J,P)$, and $C \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(K,Q)$ such that $I = J \cup K$. Then*

$$A \sqsubseteq B \bowtie C \Rightarrow A \sqsubseteq B \bowtie \mathbb{O}_{I\setminus J,\emptyset} \wedge A \sqsubseteq C \bowtie \mathbb{O}_{I\setminus K,\emptyset} \ ,$$

provided $P \cap Q = \emptyset$. The converse implication holds if, moreover, A is deterministic.

4 Operator Comparison

In this section we compare mix composition to a number of composition operators studied in the literature. Mix composition generalises the parallel composition operators, $\|$ and $\times$ defined, e.g., by Kawamoto et al. [16]. We rephrase their definition, adapted to our channels.

Definition 23. *Given $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,O)$, $B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,P)$, and $C \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(J,P)$ with $I \cap J = O \cap P = \emptyset$ define the*

- parallel composition with shared inputs $A\|B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I, O \cup P)$ *of A and B by $(A\|B)_{x,y} = A_{x,y\downarrow_O} B_{x,y\downarrow_P}$, and*
- *the* parallel composition (with distinct inputs) $A \times C \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I \cup J, O \cup P)$ *of A and C by $(A \times C)_{x,z} = A_{x\downarrow_I, z\downarrow_O} C_{x\downarrow_J, z\downarrow_P}$.*

From this definition it is obvious that we have

Corollary 24. – *Parallel composition with shared inputs $\|$ is $\bowtie$ restricted to channels with the same input names and disjoint output names.*
- *Parallel composition (with distinct inputs) $\times$ is $\bowtie$ restricted to channels with disjoint input names and disjoint output names.*

Oftentimes, the operators $\|$ and $\times$ are sufficient and more convenient to use than $\bowtie$. Technically, they always are sufficient unless outputs are fused, as we show next.

Proposition 25. *If A and B have disjoint output names then*

$$A \bowtie B = (A \times \mathbb{O}_{J\setminus I,\emptyset}) \| (B \times \mathbb{O}_{I\setminus J,\emptyset}) \ .$$

The results proved for $\bowtie$ above specialise to the following.

Corollary 26. *Let $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,O)$, $B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,P)$, $C \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,Q)$, $D \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(J,R)$, $E \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(K,S)$ such that I, J, K, O, P, Q, and S are pair-wise disjoint.*

$$\begin{aligned} &A \equiv A\|\mathbb{O}_{I,\emptyset} && A \equiv A \times \mathbb{O}_{\emptyset,\emptyset} \\ &A\|B \equiv B\|A && A \times D \equiv D \times A \\ &(A\|B)\|C \equiv A\|(B\|C) && (A \times D) \times E \equiv A \times (D \times E) \\ &A\|B \sqsubseteq A && A \times D \sqsubseteq A \times \mathbb{O}_{J,\emptyset} \end{aligned}$$

$$\begin{aligned} A \sqsubseteq A' \wedge B \sqsubseteq B' &\Rightarrow A\|B \sqsubseteq A'\|B' \\ A \sqsubseteq A' \wedge D \sqsubseteq D' &\Rightarrow A \times D \sqsubseteq A' \times D' \\ A \sqsubseteq A_1\|A_2 &\Rightarrow A \sqsubseteq A_1 \wedge A \sqsubseteq A_2 \\ A \sqsubseteq D_1 \times D_2 &\Rightarrow A \sqsubseteq D_1 \times \mathbb{O}_{I\setminus X,\emptyset} \wedge A \sqsubseteq D_2 \times \mathbb{O}_{I\setminus Y,\emptyset} \end{aligned}$$

If A is also deterministic we have:

$$\begin{aligned} A \sqsubseteq A_1 \wedge A \sqsubseteq A_2 &\Rightarrow A \sqsubseteq A_1\|A_2 \\ A \sqsubseteq D_1 \times \mathbb{O}_{I\setminus X,\emptyset} \wedge A \sqsubseteq D_2 \times \mathbb{O}_{I\setminus Y,\emptyset} &\Rightarrow A \sqsubseteq D_1 \times D_2 \end{aligned}$$

While mix composition subsumes the two parallel composition operators, $\|$ and $\times$, there are compositions that cannot be expressed with $\bowtie$ alone. The obvious example is sequential composition. But those two together are rather powerful.

A first example is the non-standard sequential composition operator defined by Barthe and Köpf [3] called *adaptive composition* by Espinoza and Smith [10]. It differs from the usual sequential composition in that the second component receives not only the output but also the input of the first as input.

Definition 27. *Let $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I, M)$ and $B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I \cup M, O)$. Provided $I \cap M = \emptyset$, define the* adaptive composition *$A \triangleright B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I, O)$ by $(A \triangleright B)_{i,o} = \sum_{m \in d(M)} A_{i,m} B_{i \cup m,o}$, for all $i \in d(I)$ and $o \in d(O)$.*

Another operator mentioned in [10] models repeated independent runs of a channel. To prevent the copies of the channel from colluding we need to disambiguate their output names with distinct tags, e.g., numbers.

Definition 28. *Let $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I, O)$ and $n \in \mathbb{N}$ such that $(i, o) \in \mathcal{V}$ and $X_{(i,o)} = X_o$, for all $i \in 1..n$ and $o \in O$.*

Define the n repeated independent runs of *A channel $A^{(n)} \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I, 1..n \times O)$ by $(A^{(n)})_{x,y} = \prod_{i=1}^{n} A_{x,\lambda o:O.y(i,o)}$, for all $x \in d(I)$ and $y \in d(1..n \times O)$.*

Adaptive composition can be expressed using ";", "$\|$" and identity channels. To express n repeated independent runs we require n renaming channels to disambiguate the copies of the output names.

Proposition 29. *$A \triangleright B = (\mathbb{I}_I \| A); B$ and $A^{(n)} = \|_{i=1}^{n} (A; R_{O,\{i\} \times O})$.*

5 Related Work

In their seminal paper Goguen and Meseguer lamented that

> Most of the models given in the literature [. . .] are not mathematically rigorous enough to support meaningful determinations of the kind needed; some do not support a sufficiently general view of security (for example, they may fail to handle breaches of security by cooperating multiple users). [12, p. 12]

We argue that $\bowtie$ is better at modelling colluding adversaries by allowing selectively shared inputs and outputs—a feature absent in the usual definitions of $\|$ and $\times$.

Gray and Syverson [13] extended with temporal operators the epistemic logic with probabilities of Halpern and Tuttle [15] to lay the foundation for a rigorous analysis of probabilistic channels. Their work is however concerned only with perfect security, that is, no leakage whatsoever.

In possibilistic settings, some recent works have presented preliminary findings for notions of refinement that preserve information-flow security properties [20, 26]. For probabilistic systems McIver et al. [25] present rely-guarantee rules.

Kawamoto et al. [16] explain how to decompose channels using $\|$ and $\times$ to then compute upper and lower bounds on measures of leakage such as g-leakage and min-entropy from the corresponding measures of the component channels. At the time of writing, the most recent version of this paper [17] mentions a connection to refinement including our Theorem 19 albeit without proof and based on a different (faulty) definition of $\sqsubseteq$.

The abstract channels as introduced by McIver et al. [24] are too abstract for our purposes. After abstracting from the names of outputs, we can no longer model fused outputs as we did, e.g., when describing two colluding neighbours in the dining cryptographers example. The programs considered in [22,23] lack any form of parallel composition although $\|$ is defined and discussed in the appendix of the latter.

All these concurrent composition operators resemble the *distributed knowledge* of two agents observing different channels as described e.g. in [11] but in a probabilistic setting. The literature on knowledge in probabilistic worlds however appears to have gone in different research directions. Halpern and O'Neill [14] characterised notions of perfect secrecy for various classes of systems including ones with probabilistic choice. Clarkson et al. [8,9] incorporate how an attacker's beliefs can change over time while observing intermediate outputs.

6 Future Work and Conclusion

Future directions for this line of work include:

- investigating further the role of collusion, that is, common output names. So far these clashes are typically either a nuisance or a triviality. Do they make for a more powerful or more elegant algebra similar to how predicate transformers that ignore Dijkstra's healthiness conditions make for a cleaner refinement algebra of sequential programs?
- exploring the concept of channel algebra further. Our channel model and $\bowtie$ composition may be steps in the right direction but are these the only necessary ingredients?
- finding bounds on various leakage measures for $\bowtie$ compositions similar to the results in [16] for $\|$ and $\times$.
- lifting channel algebra to the level of a programming language, resulting in leakage-sensitive refinement laws for programs.
- mechanising channel algebra in a theorem prover to facilitate evaluation on less trivial examples. We wrote a simple implementation of channels and operations on them, and used it for all our examples, but this library is not yet hooked up with a theorem prover for algebraic reasoning. The companion project for possibilistic compositional refinement is much more progressed in that respect [26]. Some of the infrastructure of that project could be recycled for the qif version.
- investigating how stages of verified compilers such as CompCert [19] and CakeML [18] affect leakage and how to enforce leakage bound preservation by compilation with the help of code transformations [1,4].

We feel that we have so far only scratched the surface of the possibilities opened up by the slight change of channel model and the addition of the $\bowtie$ operator. The latter appears to be a better parallel composition operator, generalising all existing ones and allowing for selective sharing, compared to the all-or-nothing of $\|$ and $\times$. This paper attempts to make a case for adopting the channel model and the $\bowtie$ operator, thereby expressing little more than the author's preferences. To the best of our knowledge, some of the results are new, including Theorem 22, or correctly stated and proved for the first time in this generality, such as Theorem 19. Besides the dining cryptographers, we have analysed a few more examples such as the combined leakage of two C bit masking assignments, all of which benefit from the new model and $\bowtie$.

Acknowledgement. For helpful discussions and comments on preliminary versions of this paper I would like to thank Carroll Morgan and Ron van der Meyden. I thank the anonymous referees for their detailed and most useful comments.

A Proofs

Proof (of Proposition 5). Let $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,O)$ and $B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(J,P)$ be $\bowtie$-compatible.

1. For well-definedness of $A \bowtie B$ it suffices to see that the denominator $D_x = \sum_{z \in \mathrm{d}(O \cup P)} A_{x\downarrow_I, z\downarrow_O} B_{x\downarrow_J, z\downarrow_P}$ is non-zero, for all $x \in \mathrm{d}(I \cup J)$. It is then clear that it normalises each row vector to sum one. Let $x \in \mathrm{d}(I \cup J)$. For the denominator to be zero it is required that $A_{x\downarrow_I, z\downarrow_O} B_{x\downarrow_J, z\downarrow_P} = 0$, for all $z \in \mathrm{d}(O \cup P)$. But that contradicts our assumption of $\bowtie$-compatibility.
   Commutativity and associativity of $\bowtie$ follow from the same properties of multiplication.
2. If A and B are both deterministic then there's exactly one 1 in each of their rows, which, together with $\bowtie$-compatibility implies that there is exactly one $z \in \mathrm{d}(O \cup P)$ for which $A_{x\downarrow_I, z\downarrow_O} = B_{x\downarrow_J, z\downarrow_P} = 1$. Whence $A \bowtie B$ is also deterministic.
3. Each channel is $\bowtie$-compatible with itself. If A is also deterministic, then we have $A_{x,y} = A^2_{x,y} = (A \bowtie A)_{x,y}$. □

Proof (of Proposition 7). If there are two different non-zero cells in a row of A, then the smaller one will decrease in $A \bowtie A$ by the normalisation involved. On the other hand, if there's just one such non-zero value, then the normalisation has no effect on that row.

□

Proof (of Proposition 8). This follows on from the observation in the previous proof. The limit must satisfy $A^{(\infty)} \bowtie A^{(\infty)} = A^{(\infty)}$.

□

Proof (of Proposition 9). Let $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,O)$ and $B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(J,P)$. For the first claim, suppose A and B are deterministic and $\bowtie$-compatible. As per Proposition 5. 2, $A \bowtie B$ is deterministic, too. This implies that row normalisation is not required.

Finally, in case A's and B's output names are disjoint, i.e., $O \cap P = \emptyset$ holds (*), we check that A and B are $\bowtie$-compatible and that the denominator is one whenever the row sums of A and B are.

$$\begin{aligned}
\sum_{z \in \mathrm{d}(O \cup P)} A_{x\downarrow_I, z\downarrow_O} \cdot B_{x\downarrow_J, z\downarrow_P} &\overset{(*)}{=} \sum_{c \in \mathrm{d}(O)} \sum_{d \in \mathrm{d}(P)} A_{x\downarrow_I, c} \cdot B_{x\downarrow_J, d} \\
&= \sum_{c \in \mathrm{d}(O)} \left(A_{x\downarrow_I, c} \sum_{d \in \mathrm{d}(P)} B_{x\downarrow_J, d} \right) \\
&= \sum_{c \in \mathrm{d}(O)} A_{x\downarrow_I, c} \qquad = 1
\end{aligned}$$
□

Proof (of Proposition 10). By associativity and commutativity of $\bowtie$, as well as idempotence on deterministic channels, we have that $A \bowtie (B \bowtie C) = A \bowtie A \bowtie B \bowtie C = (A \bowtie B) \bowtie (A \bowtie C)$. □

Proof (of Proposition 12). Let $x, y \in \mathrm{d}(I \cup J)$.

$$\begin{aligned}
(\mathbb{I}_I \bowtie \mathbb{I}_J)_{x,y} &= (\mathbb{I}_I)_{x\downarrow_I, y\downarrow_I} (\mathbb{I}_J)_{x\downarrow_J, y\downarrow_J} \\
&= \delta_{x\downarrow_I, y\downarrow_I} \delta_{x\downarrow_J, y\downarrow_J} \\
&= \delta_{x\downarrow_I \cup x\downarrow_J, y\downarrow_I \cup y\downarrow_J} \\
&= \delta_{x,y} \qquad = (\mathbb{I}_{I \cup J})_{x,y}
\end{aligned}$$
□

Proof (of Proposition 18). Each proof requires finding one or two post-processing channels. We provide them in the following table.

claim	$\sqsubseteq$	$\sqsupseteq$
(1)	A	
(2)	$\mathbb{O}_{O,Q}$	
(3)	$\mathbb{I}_I \bowtie A$	$\mathbb{I}_I \bowtie \mathbb{O}_{O \setminus I, \emptyset}$
(4)	$\mathbb{I}_O \bowtie \mathbb{O}_{Q \setminus O, \emptyset}$	$\mathbb{I}_O \bowtie \mathbb{O}_{\emptyset, Q \setminus O}$

E.g., for the "$\sqsubseteq$"-direction of claim (3), we propose to use the post-processing channel $\mathbb{I}_I \bowtie A$, that is, we claim that $\mathbb{I}_I ; (\mathbb{I}_I \bowtie A) = \mathbb{I}_I \bowtie A$ and hence $\mathbb{I}_I \sqsubseteq \mathbb{I}_I \bowtie A$. □

Proof (of Theorem 19). Let $I, J, O, P, O', P' \subseteq \mathcal{V}$ such that $O \cap P = O' \cap P' = \emptyset$. Let $A \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,O)$, $A' \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(I,O')$, $B \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(J,P)$, and $B' \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(J,P')$ such that $A \sqsubseteq A'$ and $B \sqsubseteq B'$. Let $D \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(O,O')$ and $E \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(P,P')$ such that $A; D = A'$ and $B; E = B'$. Let $x \in \mathrm{d}(I \cup J)$ and $z \in \mathrm{d}(O' \cup P')$. We show that $((A \bowtie B); (D \bowtie E))_{x,z} = (A' \bowtie B')_{x,z}$. Note that, by Proposition 5, none of the three mix compositions requires row normalisation.

$$((A \bowtie B); (D \bowtie E))_{x,z} = \sum_{y \in \mathrm{d}(O \cup P)} (A \bowtie B)_{x,y} (D \bowtie E)_{y,z}$$

$$
\begin{aligned}
&= \sum_{a\in \mathrm{d}(O)} \sum_{b\in \mathrm{d}(P)} A_{x\downarrow_I,a} B_{x\downarrow_J,b} D_{a,z\downarrow_{O'}} E_{b,z\downarrow_{P'}} \\
&= \sum_{a\in \mathrm{d}(O)} \sum_{b\in \mathrm{d}(P)} A_{x\downarrow_I,a} D_{a,z\downarrow_{O'}} B_{x\downarrow_J,b} E_{b,z\downarrow_{P'}} \\
&= \sum_{a\in \mathrm{d}(O)} A_{x\downarrow_I,a} D_{a,z\downarrow_{O'}} \cdot \sum_{b\in \mathrm{d}(P)} B_{x\downarrow_J,b} E_{b,z\downarrow_{P'}} \\
&= (AD)_{x\downarrow_I,z\downarrow_{O'}} \cdot (BE)_{x\downarrow_J,z\downarrow_{P'}} \\
&= A'_{x\downarrow_I,z\downarrow_{O'}} \cdot B'_{x\downarrow_J,z\downarrow_{P'}} \qquad = (A' \bowtie B')_{x,z}
\end{aligned}
$$

We conclude that $(A \bowtie B);(D \bowtie E) = A' \bowtie B'$ and hence $A \bowtie B \sqsubseteq A' \bowtie B'$.□

Proof (of Theorem 22). "⇒:" follows by two applications of Theorem 19 once we realise that $B \bowtie \mathbb{O}_{I\setminus J,\emptyset} = B \bowtie \mathbb{O}_{K,\emptyset}$ and $C \bowtie \mathbb{O}_{I\setminus K,\emptyset} = C \bowtie \mathbb{O}_{J,\emptyset}$.

"⇐:" For the converse, suppose A is deterministic, $A \sqsubseteq B \bowtie \mathbb{O}_{I\setminus J,\emptyset}$, and $A \sqsubseteq C \bowtie \mathbb{O}_{I\setminus K,\emptyset}$. Let $E \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(O,P)$ and $F \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(O,Q)$ satisfy $A;E = B \bowtie \mathbb{O}_{I\setminus J,\emptyset}$ and $A;F = C \bowtie \mathbb{O}_{I\setminus K,\emptyset}$. Define $D \in \mathcal{C}_{\mathcal{V},\mathcal{X}}(O,P\cup Q)$ by $D_{o,z} = E_{o,z\downarrow_P} F_{o,z\downarrow_Q}$. Let $x \in \mathrm{d}(I)$; let $z \in\in \mathrm{d}(P\cup Q)$. Define $p = z\downarrow_P$ and $q = z\downarrow_Q$. We show that $(A;D)_{x,z} = (B \bowtie C)_{x,z}$.

$$
(A;D)_{x,z} = \sum_{o\in \mathrm{d}(O)} A_{x,o} D_{o,z} = \sum_{o\in \mathrm{d}(O)} A_{x,o} E_{o,z\downarrow_P} F_{o,z\downarrow_Q}
$$

using that $A_{x,o} = A^2_{x,o}$, which follows from $A_{x,o} \in \{0,1\}$:

$$
= \sum_{o\in \mathrm{d}(O)} A^2_{x,o} \cdot E_{o,z\downarrow_P} F_{o,z\downarrow_Q}
$$

using that $A_{x,o} = 0$ or $A_{x,o'} = 0$ for all $o' \neq o$:

$$
\begin{aligned}
&= \sum_{o\in \mathrm{d}(O)} A_{x,o} \sum_{o'\in \mathrm{d}(O)} A_{x,o'} E_{o,z\downarrow_P} F_{o',z\downarrow_Q} \\
&= \left(\sum_{o\in \mathrm{d}(O)} A_{x,o} E_{o,z\downarrow_P} \right) \sum_{o\in \mathrm{d}(O)} A_{x,o} F_{o,z\downarrow_Q} \\
&= (A;E)_{x,z\downarrow_P} \cdot (A;F)_{x,z\downarrow_Q} \\
&= (B \bowtie \mathbb{O}_{I\setminus J,\emptyset})_{x,z\downarrow_P} \cdot (C \bowtie \mathbb{O}_{I\setminus K,\emptyset})_{x,z\downarrow_Q} \\
&= B_{x\downarrow_J,z\downarrow_P} \cdot (\mathbb{O}_{I\setminus J,\emptyset})_{x\downarrow_{(I\setminus J)},\emptyset} \cdot C_{x\downarrow_K,z\downarrow_Q} \cdot (\mathbb{O}_{I\setminus K,\emptyset})_{x\downarrow_{(I\setminus K)},\emptyset} \\
&= B_{x\downarrow_J,z\downarrow_P} \cdot C_{x\downarrow_K,z\downarrow_Q} \qquad = (B \bowtie C)_{x,z}
\end{aligned}
$$

It follows that $A;D = B \bowtie C$ and hence $A \sqsubseteq B \bowtie C$. □

Proof (of Proposition 25).

$$
(A \bowtie B)_{x,y} = A_{x\downarrow_I,y\downarrow_O} \cdot B_{x\downarrow_J,y\downarrow_P}
$$

$$\begin{aligned}
&= A_{x\downarrow_I, y\downarrow_O} \cdot (\mathbb{O}_{J\setminus I,\emptyset})_{x\downarrow_{(J\setminus I)},\emptyset} \cdot B_{x\downarrow_J, y\downarrow_P} \cdot (\mathbb{O}_{J\setminus I,\emptyset})_{x\downarrow_{(I\setminus J)},\emptyset} \\
&= (A \times \mathbb{O}_{J\setminus I,\emptyset})_{x,y\downarrow_O} \cdot (B \times \mathbb{O}_{J\setminus I,\emptyset})_{x,y\downarrow_P} \\
&= ((A \times \mathbb{O}_{J\setminus I,\emptyset}) \| (B \times \mathbb{O}_{J\setminus I,\emptyset}))_{x,y} \qquad \square
\end{aligned}$$

Proof (of Proposition 29). First let $x \in \mathrm{d}(I)$ and $y \in \mathrm{d}(O)$.

$$\begin{aligned}
(A \triangleright B)_{x,y} &= \sum_{m \in \mathrm{d}(M)} A_{x,m} B_{x \cup m, y} \\
&= \sum_{m \in \mathrm{d}(M)} A_{x,m} B_{x \cup m, y} \\
&= \sum_{m' \in \mathrm{d}(I \cup M)} \delta_{x, m'\downarrow_I} A_{x, m'\downarrow_O} B_{m',y} \\
&= \sum_{m' \in \mathrm{d}(I \cup M)} (\mathbb{I}_I)_{x, m'\downarrow_I} A_{x, m'\downarrow_O} B_{m',y} \\
&= \sum_{m' \in \mathrm{d}(I \cup M)} (\mathbb{I}_I \| A)_{x,m'} B_{m',y} \qquad = ((\mathbb{I}_I \| A); B)_{x,y}
\end{aligned}$$

Now let $y \in \mathrm{d}(1..n \times O)$.

$$\begin{aligned}
(A^{(n)})_{x,y} &= \prod_{i=1}^{n} A_{x, \lambda o:O.y(i,o)} \\
&= \prod_{i=1}^{n} \sum_{m \in \mathrm{d}(O)} A_{x,m} \delta_{m, \lambda o:O.y(i,o)} \\
&= \prod_{i=1}^{n} \sum_{m \in \mathrm{d}(O)} A_{x,m} (\mathrm{R}_{O,\{i\}\times O})_{m, y\downarrow_{\{i\}\times O}} \\
&= \prod_{i=1}^{n} (A; \mathrm{R}_{O,\{i\}\times O})_{x, y\downarrow_{\{i\}\times O}} \qquad = \|_{i=1}^{n} (A; \mathrm{R}_{O,\{i\}\times O})_{x,y} \qquad \square
\end{aligned}$$

References

1. Agat, J.: Transforming out timing leaks. In: Wegman, M.N., Reps, T.W. (eds.) POPL 2000, Proceedings of the 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Boston, Massachusetts, USA, 19–21 January, 2000, pp. 40–53. ACM (2000), http://doi.acm.org/10.1145/325694.325702
2. Alvim, M.S., Chatzikokolakis, K., Palamidessi, C., Smith, G.: Measuring information leakage using generalized gain functions. In: Chong, S. (ed.) 25th IEEE Computer Security Foundations Symposium, CSF 2012, Cambridge, MA, USA, 25–27 June 2012, pp. 265–279. IEEE Computer Society (2012), http://dx.doi.org/10.1109/CSF.2012.26

3. Barthe, G., Köpf, B.: Information-theoretic bounds for differentially private mechanisms. In: Proceedings of the 2011 IEEE 24th Computer Security Foundations Symposium, CSF 2011, pp. 191–204 (2011), http://dx.doi.org/10.1109/CSF.2011.20
4. Barthe, G., Rezk, T., Warnier, M.: Preventing timing leaks through transactional branching instructions. In: Cerone, A., Wiklicky, H. (eds.) Proceedings of the Third Workshop on Quantitative Aspects of Programming Languages (QAPL 2005). ENTCS, vol. 153(2), pp. 33–55 (2006), https://doi.org/10.1016/j.entcs.2005.10.031
5. Blackwell, D.: Comparison of experiments. In: Neyman, J. (ed.) Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, pp. 93–102. Univ. of Calif. Press (1951), http://projecteuclid.org/euclid.bsmsp/1200500222
6. Braun, C., Chatzikokolakis, K., Palamidessi, C.: Quantitative notions of leakage for one-try attacks. In: Proceedings of the 25th Conference on Mathematical Foundations of Programming Semantics (MFPS 2009). ENTCS, vol. 249, pp. 75–91 (2009), http://dx.doi.org/10.1016/j.entcs.2009.07.085
7. Chaum, D.: The dining cryptographers problem: unconditional sender and recipient untraceability. J. Crypto. **1**(1), 65–75 (1988)
8. Clarkson, M.R., Myers, A.C., Schneider, F.B.: Belief in information flow. In: 18th IEEE Computer Security Foundations Workshop, (CSFW-18 2005), 20–22, Aix-en-Provence, France, pp. 31–45. IEEE Computer Society (2005), http://dx.doi.org/10.1109/CSFW.2005.10
9. Clarkson, M.R., Myers, A.C., Schneider, F.B.: Quantifying information flow with beliefs. J. Comput. Secur. **17**(5), 655–701 (2009), http://dx.doi.org/10.3233/JCS-2009-0353
10. Espinoza, B., Smith, G.: Min-entropy as a resource. Inf. Comput., **226**, 57–75 (2013). Blakey, Coecke, B., Mislove, M., Pavlovic, D.: Information Security as a Resource (special Issue), http://dx.doi.org/10.1016/j.ic.2013.03.005
11. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning About Knowledge. MIT-Press (1995)
12. Goguen, J.A., Meseguer, J.: Security policies and security models. In: 1982 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 26–28 April 1982, pp. 11–20 (1982), http://ieeexplore.ieee.org/document/6234468
13. Gray III., J.W., Syverson, P.F.: A logical approach to multilevel security of probabilistic systems. Distrib. Comput. **11**(2), 73–90 (1998), http://dx.doi.org/10.1007/s004460050043
14. Halpern, J.Y., O'Neill, K.R.: Secrecy in multiagent systems. ACM Trans. Inf. Syst. Secur. (TISSEC) **12**(1), 5 (2008)
15. Halpern, J.Y., Tuttle, M.R.: Knowledge, probability, and adversaries. J. ACM **40**(4), 917–960 (1993), http://doi.acm.org/10.1145/153724.153770
16. Kawamoto, Y., Chatzikokolakis, K., Palamidessi, C.: Compositionality results for quantitative information flow. In: Norman, G., Sanders, W. (eds.) QEST 2014. LNCS, vol. 8657, pp. 368–383. Springer, Cham (2014). doi:10.1007/978-3-319-10696-0_28
17. Kawamoto, Y., Chatzikokolakis, K., Palamidessi, C.: On the compositionality of quantitative information flow. CoRR abs/1611.00455 (2016), http://arxiv.org/abs/1611.00455

18. Kumar, R., Myreen, M.O., Norrish, M., Owens, S.: CakeML: a verified implementation of ML. In: Jagannathan, S., Sewell, P. (eds.) The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2014, San Diego, CA, USA, 20–21 January 2014, pp. 179–192. ACM (2014), http://doi.acm.org/10.1145/2535838.2535841
19. Leroy, X.: Formal verification of a realistic compiler. Commun. ACM **52**(7), 107–115 (2009), http://doi.acm.org/10.1145/1538788.1538814
20. Mantel, H.: Preserving information flow properties under refinement. In: 2001 IEEE Symposium on Security and Privacy, Oakland, California, USA, 14–16 May 2001, pp. 78–91. IEEE Computer Society (2001), http://dx.doi.org/10.1109/SECPRI.2001.924289
21. McIver, A., Meinicke, L., Morgan, C.: Compositional closure for bayes risk in probabilistic noninterference. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP 2010. LNCS, vol. 6199, pp. 223–235. Springer, Heidelberg (2010). doi:10.1007/978-3-642-14162-1_19
22. McIver, A., Meinicke, L., Morgan, C.: Hidden-Markov program algebra with iteration. Math. Struct. Comput. Sci. **25**(2), 320–360 (2015), https://doi.org/10.1017/S0960129513000625
23. McIver, A., Morgan, C., Rabehaja, T.M.: Abstract hidden Markov models: a monadic account of quantitative information flow. In: 30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, 6–10 July 2015, pp. 597–608. IEEE Computer Society (2015), https://doi.org/10.1109/LICS.2015.61
24. McIver, A., Morgan, C., Smith, G., Espinoza, B., Meinicke, L.: Abstract channels and their robust information-leakage ordering. In: Abadi, M., Kremer, S. (eds.) POST 2014. LNCS, vol. 8414, pp. 83–102. Springer, Heidelberg (2014). doi:10.1007/978-3-642-54792-8_5
25. McIver, A., Rabehaja, T.M., Struth, G.: Probabilistic rely-guarantee calculus (v3). CoRR abs/1409.0582 (2015), http://arxiv.org/abs/1409.0582
26. Murray, T.C., Sison, R., Pierzchalski, E., Rizkallah, C.: Compositional verification and refinement of concurrent value-dependent noninterference. In: IEEE 29th Computer Security Foundations Symposium, CSF 2016, Lisbon, Portugal, 27 June–1 July 2016, pp. 417–431. IEEE Computer Society (2016), http://dx.doi.org/10.1109/CSF.2016.36
27. Smith, G.: Recent developments in quantitative information flow (invited tutorial). In: Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), LICS 2015, pp. 23–31 (2015), http://dx.doi.org/10.1109/LICS.2015.13