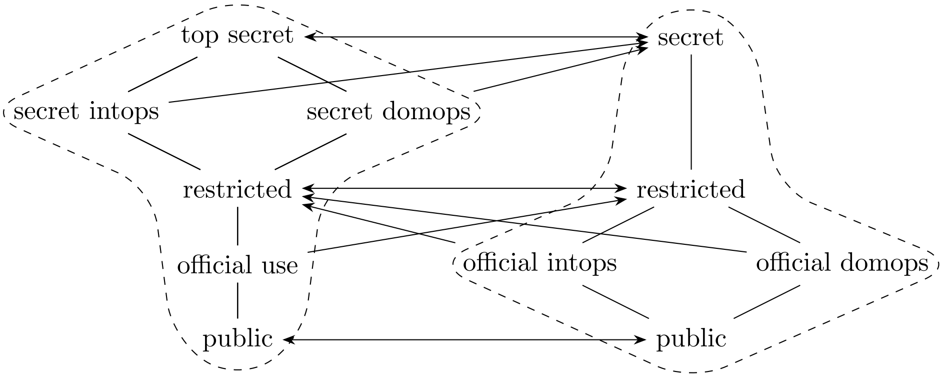
A Coq Formalization of Lagois Connections for Secure Information Flow

Casper Ståhl, Levs Gondelmans,
René Rydhof Hansen, and Danny Bøgsted Poulsen

Aalborg University, Aalborg, Denmark
cstahl20@student.aau.dk, {lego,rrh,dannybpoulsen}@cs.aau.dk

Consider the situation illustrated below, in which two organisations wish to communicate in a secure way without compromising their respective local information flow policies, as defined using *security lattices* [4]. This becomes a matter of figuring out how to map security levels in one organisation to security levels in the other organisation and vice versa, while taking the intended communication pattern into account to ensure that no information can be leaked in one organisation by sending it on a "roundtrip" through the other organisation:



While the local security policies can be enforced, e.g., by type systems [6] or static analysis [3], it is non-trivial to prove that a given mapping between security levels is secure, i.e., that the local information security policies are not compromised even when taking external communication into account. Bhardwaj and Prasad propose the use of *Lagois connections* (as defined by Melton [5]) as a framework defining such mappings in a way that is secure by design, yet without mapping security levels to unnecessarily high counterparts [1, 2] and thereby avoid "label creep". Lagois connections are very similar to the more commonly known Galois connections with a slight change:

Definition 1 (Lagois connection)**.** A poset system (P, f, g, Q) is called an *(increasing) Lagois connection* iff $p \leq gf(p)$ for all $p : P$ (**LC1**), $q \leq fg(q)$ for all $q : Q$ (**LC2**), $fgf(p) = f(p)$ for all $p : P$ (**LC3**), and $gfg(q) = g(q)$ for all $q : Q$ (**LC4**).

With this definition in mind, flow is permitted between the two systems from $p\colon P$ to $q\colon Q$ when $f(p) \leq q$ and analogously the other way around. As an example, in the figure above information is allowed to flow from "secret intops" to "top secret" via the lattice itself but also by a round trip of the Lagois connection via "secret". The fact that f and g are monotone, by virtue of (P, f, g, Q) being a poset system, ensures *one way security* in the sense that policies are preserved post mapping, or more precisely $p \leq p'$ implies $f(p) \leq f(p')$ for all $p\, p' : P$ (analogous for g). **LC1** and **LC2** ensures back and forth security, in the sense that mapping a label $p : P$ back and forth resulting in $gf(p)$, respects the local policy in the sense that $p \leq gf(p)$ (analogous for fg). In relation to information flow, **LC3** and **LC4** mainly ensure that the mappings are *precise* and that continued back and forth communication immediately converges after one round trip.

Our contribution is a Coq formalization of Lagois connections and related theory for use in secure information flow[1]. In particular, we have formalised the results from the seminal work of Melton et al. [5] that are of interest for secure information flow [1]. Further, we have used this formalisation to develop a formal variation of the type system developed by Bhardwaj and Prasad [1, 2] and have proved, formally in Coq, that that our variation is sound with respect to non-interference. Since our primary interest is secure information flow, the formalisation is limited to *security lattices*, that is, finite inhabited lattices, allowing for a more direct representation of many results by Melton et al. More importantly this moves us in a direction where Lagois connections can automatically be established and proved within Coq.

Our main result is a formal proof of soundness for a type system very similar to the one presented by Bhardwaj and Prasad [1, 2], i.e., one that is better suited for formal proofs and conjectured to be equivalent:

Theorem 1. *For a Lagois connection (P, f, g, Q), and an adversarial residing at level $p : P$ and $q : Q$ such that $p = g(q)$ and $q = f(q)$. If for a program s, environments ν, ν_f, ν', ν'_f and μ, μ_f, μ', μ'_f belonging to the organizations of P and Q respectively:*

1. *starting in environment (ν, μ), executing s evaluates to (ν_f, μ_f),*
2. *starting in environment (ν', μ'), executing s evaluates to (ν'_f, μ'_f),*
3. *s can be given security type (p', q'), and*
4. *$\nu(v) = \nu'(v)$ for all v with a security type $p'' : P$ such $p'' \leq p$ and $\mu(v) = \mu'(v)$ for all w with a security type $q'' : Q$ such $q'' \leq q$.*

Then $\nu_f(v) = \nu'_f(v)$ for all v with a security type $p'' : P$ such $p'' \leq p$ and $\mu_f(v) = \mu'_f(v)$ for all w with security type $q'' : Q$ such $q'' \leq q$.

Intuitively, this result states that for two systems communicating over channels that respect an underlying Lagois connection, *non-interference* will be guaranteed in each system, even taking the communication into account. I.e., no information is leaked, even if has been on a "roundtrip" through the other system. Essential to proving this result, the type system additionally checks if the Lagois connection is respected when communication takes place.

In addition to providing a formalisation of existing work, our approach allowed us to write certain proofs in a more direct style, simplifying working with both proofs and formalisation. Furthermore, we argue that our formalisation is a solid foundation for further work on both Lagois connections and the use of these for secure information flow. This is exemplified in our current work, using (and extending) the framework to investigate and formalise secure information flow between more than two systems as current methods [2] for extending the framework do not do so nicely: Composing Lagois connections $(P, f, g, Q), (Q, \hat{f}, \hat{g}, R)$ in a chain as $\mathbf{L} = (P, \hat{f}f, \hat{g}g, R) = (P, f, g, Q) \circ (Q, \hat{f}, \hat{g}, R)$ is secure in the sense that $\mathbf{L}$ satisfies both **LC1** and **LC2**, but such compositions may not be precise and may not converge (quickly). For example, $\mathbf{L}$ is only a Lagois connection iff $\hat{g}\hat{f}f[P] \subseteq f[P]$ and $fg\hat{g}[R] \subseteq \hat{g}[R]$ (Theorem 3.22 in [5]). Further, we have observed chains of Lagois connections $\mathbf{L} = (P, f, g, Q) \circ (Q, \hat{f}, \hat{g}, R)$

[1] The formalisation is available at https://github.com/CasperStaahl/TYPES-2025-formalization-preview

that are lossy, in the sense that the established Lagois connection $\mathbf{L}$ is less precise than one that could be established between P and R directly even though (P, f, g, Q) and $(Q, \hat{f}, \hat{g}, R)$ are as precise as possible.

In our ongoing work we explore ways to mitigate this limitation by considering Lagois connections over communication topologies given by undirected graphs. Not all topologies "automatically" give rise to secure communications, but some do and we are currently working to characterise such topologies.

References

[1] Chandrika Bhardwaj and Sanjiva Prasad. "Only connect, securely". In: *Formal Techniques for Distributed Objects, Components, and Systems: 39th IFIP WG 6.1 International Conference, FORTE 2019, Held as Part of the 14th International Federated Conference on Distributed Computing Techniques, DisCoTec 2019, Kongens Lyngby, Denmark, June 17–21, 2019, Proceedings 39*. Springer. 2019, pp. 75–92.

[2] Chandrika Bhardwaj and Sanjiva Prasad. "Secure information flow connections". In: *Journal of Logical and Algebraic Methods in Programming* 127 (2022), p. 100761.

[3] Patrick Cousot and Radhia Cousot. "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints". In: *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*. 1977, pp. 238–252.

[4] Dorothy E Denning. "A lattice model of secure information flow". In: *Communications of the ACM* 19.5 (1976), pp. 236–243.

[5] Austin Melton, Bernd SW Schröder, and George E Strecker. "Lagois connections—a counterpart to Galois connections". In: *Theoretical Computer Science* 136.1 (1994), pp. 79–107.

[6] Dennis Volpano, Cynthia Irvine, and Geoffrey Smith. "A sound type system for secure flow analysis". In: *Journal of computer security* 4.2-3 (1996), pp. 167–187.