

**Username:** Jeanne Chua **Book:** Computer Security: Art and Science. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

---

## 16.6. Summary

Two aspects of information flow are the amount of information flowing and the way in which it flows. Given the value of one variable, entropy measures the amount of information that one can deduce about a second variable. The flow can be explicit, as in the assignment of the value of one variable to another, or implicit, as in the antecedent of a conditional statement depending on the conditional expression.

Traditionally, models of information flow policies form lattices. Should the models not form lattices, they can be embedded in lattice structures. Hence, analysis of information flow assumes a lattice model.

A compiler-based mechanism assesses the flow of information in a program with respect to a given information flow policy. The mechanism either certifies that the program meets the policy or shows that it fails to meet the policy. It has been shown that if a set of statements meet the information flow policy, their combination (using higher-level language programming constructs) meets the information flow policy.

Execution-based mechanisms check flows at runtime. Unlike compiler-based mechanisms, execution-based mechanisms either allow the flow to occur (if the flow satisfies the information flow policy) or block it (if the flow violates the policy). Classifications of information may be static or dynamic.

Two example information flow control mechanisms, the Security Pipeline Interface and the Secure Network Server Mail Guard, provide information flow controls at the system level rather than at the program and program statement levels.