6th International Workshop on Security Protocols Cambridge, UK, April 1998, Springer LNCS.

External Consistency and the Verification of Security Protocols

Simon N. Foley

Centre for Communications Systems Research,
University of Cambridge,
Cambridge CB2 3DS, UK.

Abstract. The notion of external consistency—that system state correctly reflects the real world—provides a basis for a denotational definition of integrity. We regard segregation of duties, well formed transactions, auditing, replication, MACs, and so forth, as simply implementation techniques: they define *how* to achieve this notion of integrity in an operational sense. Therefore, we argue that when a designer claims that a system is fault-tolerant, or that a protocol properly authenticates, or that a system is secure against fraud, then what the designer is actually claiming is that it is externally consistent. An advantage of taking this view is that it allows us to give a meaning to the 'security' of a system that uses a combination of these implementation techniques.

1 A Systems View

Requirements Analysis [10] typically identifies the *essential* functional requirements that define *what* the system must do. An implementation defines *how* the system operates and must take into consideration the fact that the infrastructure that is put in place to support the the requirements may be unreliable. For example, an infrastructure should include a backup and restore subsystem. While not part of the essential requirements, it is a necessary part of the implementation since the infrastructure can corrupt data.

Example 1 When a consignment arrives from a supplier (Figure 1) a clerk verifies it against its accompanying consignment note (cnote). This clerk, part of the infrastructure, uses the cnote to enter invoice details (inv) to a computer system, which in turn, sends payment (pay) directly to the supplier.

The essential requirement for this simplistic *enterprise* is that on receipt of a cnote, a pay should be generated. External Consistency (integrity) is maintained if, even in the presence of failures within the infrastructure, the enterprise can still support this requirement at it's external interface with the supplier. △

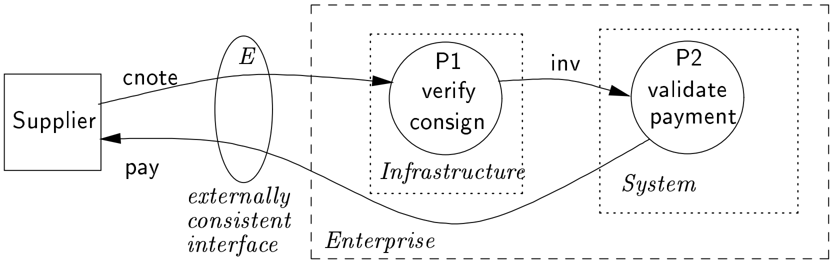

Fig. 1. Data flow diagram of a simple payment system

2 External Consistency

We assume that an enterprise is composed of a reliable and unreliable components. If external consistency is to be maintained then the reliable part of the enterprise must be capable of compensating for the unreliable section.

Suppose that the behaviour of payment enterprise is described by the composition of $\mathsf{P1}$ and $\mathsf{P2}$. Under normal operating conditions the clerk alternates between processing cnotes and invs. As a result, the $\mathsf{Supplier}$ participates in a sequence of cnote and pay messages through its interface with the enterprise. For example, the supplier could expect to engage in a sequence $\langle \mathsf{cnote}, \mathsf{pay} \rangle$ with the enterprise, but would not expect, for example, to engage $\langle \mathsf{pay}, \mathsf{pay} \rangle$.

However, it is possible that the clerk can fail and as a result take on an arbitrary behaviour $\overline{\mathsf{P1}}$, corresponding to an infrastructure failure. External consistency is achieved if the behaviour at the external interface E, resulting from an infrastructure failure, is consistent with the behaviour under normal conditions. Formally,

$$(\overline{\mathsf{P1}} || \mathsf{P2})@E \subseteq (\mathsf{P1} || \mathsf{P2})@E$$

where $\mathsf{P1}||\mathsf{P2}$ is the parallel composition of (processes) $\mathsf{P1}$ and $\mathsf{P2}$ [6] and $P@E$ is the set of observations that can be made at the interface E of process P. This is an information-flow style definition (in the sense of [1]): variety in the behaviour (normal or abnormal) of the infrastructure may not be conveyed to the interface. A formal study of this definition, and the examples in this paper may be found in [5].

3 Implementing External Consistency

It's clear that the payment subsystem, as currently described, is not externally consistent at interface E: an (abnormal) clerk can engage in arbitrary inv events, generating a stream of payments at the external interface that do not have corresponding cnote events (as per a normal behaviour).

If the system and supplier share a secret key (unknown to the clerk) then a Message Authentication Code (MAC) can be included in cnote to ensure the authenticity of the message. In this case, the abnormal behaviour of the clerk is limited in that he cannot construct arbitrary MACs. This limitation must be characterised in the specification of possible abnormal behaviour $\overline{\mathsf{P1}}$. Assuming that the design of the system $\mathsf{P2}$ is such that it is resilient to forged and replayed inv messages, then we argue that external consistency is achieved at interface E. This approach would appear to be consistent with existing protocol analysis techniques, for example, [4, 7, 8], which use a roughly similar strategy to validate authentication protocols. The protocol is specified as $\mathsf{P2}$ and the attacker (Spy) makes up the infrastructure, choosing to have normal ($\mathsf{P1}$) or abnormal ($\overline{\mathsf{P1}}$) behaviour. The key difference is that [4, 7, 8] are tailored to specific theorem-proving or model-checking environments.

An alternative (but not practical) approach to implementing integrity in the payment subsystem is to make the infrastructure fault tolerant. Assume that every consignment is processed simultaneously by $2n + 1$ replicated clerks; the system votes on the ($2n+1$ invoices) to decide whether a consignment is valid. In this case, the abnormal behaviour of the infrastructure is represented by at least $n + 1$ clerks having normal behaviour, and we argue that external consistency is achieved at the interface E. This is consistent with existing results where non-interference techniques have been used to verify fault-tolerance [9, 11].

A further alternative to implementing integrity is to use segregation of duties. Assume that a separate invoice and consignment note is used for every consignment, and these are processed by different clerks. In this case the abnormal behaviour of the infrastructure corresponds to one clerk behaving abnormally on a transaction, while the other behaves normally (clerks do not collaborate), and external consistency can be achieved.

These implementation techniques can be combined to suit the application. For example, returning to the MAC implementation described above, we did not consider how the enterprise might establish a secret key between a supplier and the system. Suppose that a supervisor is given this responsibility. So long as the supervisor (infrastructure) and the cnote-processing clerk are different people, then a failure by one cannot result in an unexpected behaviour at the external interface. This is an example of combining MACs and segregation of duties.

4 Conclusion

By considering the nature of the entire enterprise we provide a meaningful and implementation independent definition for integrity. This systems view [10] has not been adopted by conventional integrity models, such as Clark-Wilson [3], which limit themselves to the boundary of the computer system and tend to define integrity in an operational/implementation-oriented sense.

The properties used for verifying authentication in [4], and fault-tolerance in [9, 11], are similar, and are based on non-interference. Therefore, we think it reasonable to claim that our notion of external consistency captures the sentiment of what is sought when analysing authentication protocols in general[1]. However, our interest is not in verifying authentication protocols, per-se, but in verifying the external consistency of more general enterprises that use a variety of techniques to implement integrity. Our formal definition of external consistency is for illustrative purposes only. A re-formulation of this definition in terms of the existing verification approaches, such as [4, 8, 7, 9], would be preferable. Additionally, given the usefulness of BAN [2] style analysis of protocols, it would be interesting to determine whether a similar approach could be used for analysing external consistency in general.

Finally, we argued that external consistency is effectively an information flow property. There is a vast literature studying the nature of these properties (both possabilistic and probabilistic), the results of which, must be interpreted carefully. For example, preservation by composition will depend on a number of factors: the computational model, the structure of the interfaces, and so forth.

Acknowledgement

This work was done while I was a member of the CCSR, on leave from University College, Cork. I'd like to express my gratitude to Stewart Lee for inviting me and thank him, and the members of the Cambridge Computer Security Group, for a most enjoyable visit. Thanks also to Kan Zhang for discussions that helped to solidify my ideas on external consistency. This work was supported, in part, by a basic research grant from Forbairt (Ireland).

References

1. W.R. Ashby. *An Introduction to Cybernetics.* Methuen, New York, 1964.
2. M. Burrows, M. Abadi, and R. M. Needham. A logic of authentication. Technical Report Report number 39, Digital Systems Research Center, February 1989.
3. D. D. Clark and D. R. Wilson. A comparison of commercial and military computer security models. In *Proceedings Symposium on Security and Privacy*, pages 184–194. IEEE Computer Society Press, April 1987.
4. R. Focardi, A. Ghelli, and R. Gorrieri. Using noninterference for the analysis of security protocols. In *Proceedings of DIMACS Workshop on Design and Formal Verification of Security Protocols*, 1997.
5. S.N. Foley. Evaluating system integrity. April 1998. Submitted for publication.
6. C.A.R. Hoare. *Communicating Sequential Processes.* Prentice Hall, 1985.
7. L.C. Paulson. The inductive approach to verifying cryptographic protocols. In *Proceedings of the IEEE Computer Security Foundations Workshop*, 1997.
8. A.W. Roscoe. Using intensional specifications of security protocols. In *Proceedings of the IEEE Computer Security Foundations Workshop*, 1996.
9. A.C. Simpson. *Safety through Security.* PhD thesis, Oxford University, Computing Laboratory, 1996.
10. J.F. Palmer S.M. McMenamin. *Essential Systems Analysis.* Prentice Hall, 1984.
11. D. Weber. Specifications for fault-tolerance. Technical Report 19-3, Odyssey Research Associates, Ithaca, NY, 1988.

[1] At this point we do not consider confidentiality, although the authors of [4] have used non-interference to characterise confidentiality in authentication protocols.