Why Isn't Trust Transitive?

Bruce Christianson[1] and William S. Harbison[2]

[1] Faculty of Information Sciences, University of Hertfordshire : Hatfield
[2] Computer Laboratory, University of Cambridge

Abstract. One of the great strengths of public-key cryptography is its potential to allow the localization of trust. This potential is greatest when cryptography is present to guarantee data integrity rather than secrecy, and where there is no natural hierarchy of trust. Both these conditions are typically fulfilled in the commercial world, where CSCW requires sharing of data and resources across organizational boundaries. One property which trust is frequently assumed or "proved" to have is transitivity (if A trusts B and B trusts C then A trusts C) or some generalization of transitivity such as *-closure. We use the loose term *unintensional transitivity of trust* to refer to a situation where B can effectively put things into A's set of trust assumptions without A's explicit consent (or sometimes even awareness.) Any account of trust which allows such situations to arise clearly poses major obstacles to the effective confinement (localization) of trust. In this position paper, we argue against the need to accept unintensional transitivity of trust. We distinguish the notion of trust from a number of other (transitive) notions with which it is frequently confused, and argue that "proofs" of the unintensional transitivity of trust typically involve unpalatable logical assumptions as well as undesirable consequences.

1 The Need for Trust

Principals engaging in a security protocol often require justification for doing so. They must be able to prove that they are "entitled" to take the visible actions which they do on the basis of their own individual (and possibly highly private) policies. In order to prove this, they will typically need to make explicit "trust assumptions" about other parties participating in the protocol. Trust is then used as a substitute for knowledge in order to demonstrate that the protocol has the security properties that the principal desires. Theories of knowledge typically endow knowledge with a strong form of inter-subjective agreement often mis-described as "objectivity". But unlike knowledge, trust is strongly subject-dependent.

The confinement or localization of trust is desirable in order to allow a principal to contain the risk that her participation in the protocol actually endows her system with additional properties which she urgently desires it not to have.

Unwanted secret-sharing must be assumed transitive in any sensible threat model. One of the great strengths of public-key cryptography is its potential to allow the localization of trust, since it does not require any secrets to be

shared. This potential is greatest when cryptography is present to guarantee data integrity rather than their secrecy, and where there is no natural hierarchy of trust. Both these conditions are typically satisfied in the commercial world, where CSCW requires sharing of data and resources across organizational boundaries.

Trust is an elusive concept, however. Many treatments of security attempt to take trust as a primitive with postulated properties, and derive consequences. One property frequently postulated or "derived" is some form of transitivity. In the simplest case (writing "A trusts B" as shorthand for "A trusts B about X under certain conditions") we allegedly have:

```
If: A trusts B .&. B trusts C :then. A trusts C
```

frequently written with the addition of the sentiment "whether A is aware of the fact or not". We use the loose term *unintensional transitivity of trust* to refer to a situation in which B can act in such a way as to put things into A's set of trust assumptions without A's explicit consent (or sometimes even A's awareness.) Acceptance of any account of trust which allows such situations to arise clearly poses major obstacles to the effective confinement (localization) of trust.

In this position paper, we argue against accepting the unintensional transitivity of trust. We first distinguish the notion of trust from a number of other (transitive) notions with which it is frequently confused. We then argue that apparent unintensional transitivity of trust is typically an artifact of deploying unpalatable logical assumptions.

2 Some Things Which Trust is Not

Trust is not Reliance. Many statements ostensibly about trust make better sense if "trusts" is replaced by "relies upon". For example, I may be required to have my software certified by the QA department, which they do by running known tests on their hardware. Or I may be required to submit a transaction authorization which has been cryptographically signed by a particular guardian on a smart card. In both cases I rely (or depend) upon the other parties to do what is required of them. I cannot complete my part of the task unless they do theirs, and I cannot exercise any control over what they do (or don't do). But this does not mean that I trust them. I can validate all their actions, by re-running the tests on my own hardware, or by checking the signature, before I commit to my part of the transaction. I must rely upon them, but I need not trust them.

Trust is not Trustworthiness. We distinguish statements of the form "A trusts B" from statements about trustworthiness, such as:

```
A believes. B is trustworthy
```

(ie B is trustworthy as far as A is concerned, about X, under certain circumstances, etc.) The notion of belief used here is a very mild one: "A believes p" does not assert that A actually has a considered opinion on p's truth, merely that A would (be entitled to) accept and act upon p on the basis of her other beliefs, if she did think about it in a suitable monotonic logic with Modus Ponens.

The question whether B is trustworthy (relative to a particular security policy) is a question of fact. There is a reality independent of A's belief, which A might be right or wrong about. But the fact that Alice believes Bob to be trustworthy doesn't mean that she actually trusts him (she may have no need to, or may wish to spare his blushes) nor conversely: Alice may trust Bob not because she believes him to be trustworthy but because she has no choice.

Trust is an epistemic notion: statements about trust are statements about certain beliefs held by others and their reasons for holding them, not about what would make such beliefs true in the real world.

Trust is not Jurisdiction. Often we unpack

```
A believes. B has jurisdiction over X
```

by saying "A regards B as an authority on X, and part of what that means is that A should trust B about X", or

```
If. A believes: B has jurisdiction over X .&. B says X
  .:then. A believes X
```

But often this hides an ambiguity in the nature of the authority proposed. A might mean that B is an authority in the sense that X is true just because B says it. If my system manager says that my disk quota is 10Mb then my disk quota really is 10Mb. What makes it so is the fact that my system manager utters the statement (via an entry in the configuration file.) This is not why I believe it: I believe that my disk quota is 10Mb because I found out the hard way. I do not trust my system manager at all. Just because he says "your disk quota will be increased to 15Mb tomorrow" this does not make me believe it! However I do trust my friend Bob, whose visceral ability to predict such things is legendary: if Bob says that my quota is about to be increased, I am willing to bet money that it will be. Bob, of course, is (or has) a different kind of authority: his statement is the sufficient reason for my belief, rather than being what makes my belief true. The first kind of jurisdiction does not entail trust, the second kind is not something which can be unilaterally delegated or handed off. Many arguments about trust slip gently from using one kind of authority into the other, without consideration of their different natures.

Trust is not Delegation. Suppose Alice trusts Bob about X (under certain conditions which are currently fulfilled.) Suppose Bob says "I have delegated jurisdiction over X to Carol." This does not imply that Carol accepts the delegated

authority. Nor does it imply that either Alice or Bob trusts Carol to use this delegated authority correctly (although they may rely upon her to do so.) Nor does it imply that Alice accepts Carol's jurisdiction over X, unless for example Bob tells Alice that she can trust Carol, and Alice trusts Bob to tell her who to trust, which begs the question. If all operations could be delegated by unilateral decision of the delegator, then many security policies would be easy to break (eg two bank managers holding different coloured keys, delegating to the same secretary.) Trusting B not to delegate inappropriately again begs the question.

Trust may involve Getting Sacked. Another way of unpacking trust is that "A trusts B" means that B has the ability to violate A's security policy, often put loosely by saying that B has the power to get A sacked. However this ability need not be transitive: it may be that B can break A's policy and C can break B's but, because A's policy is weaker (more permissive) than B's in some crucial respect, C cannot break A's policy without B's help.

3 An Anatomy of Trust

For the sake of further argument, let's unpack "A trusts B" as:

```
if. A believes. B says X :then. A believes X
```

where we are unpacking belief here in the same way as in the previous section.

We are not particularly advocating this (or any other) specific account of trust, although we do ask you to accept that as far as reasoning about transitivity is concerned, this definition is at least as good as most others. Now assume that A believes:

```
A trusts B .&. B trusts C .&. C says X
```

and try to prove that

```
A believes X
```

Clearly this requires some further internal structure to trust. On the same rather arbitrary basis, we may choose to decompose "A trusts B" into the conjunction of "trust in honesty":

```
if. A believes. B says X :then. A believes. B believes X
```

and "trust in competence":

```
if. A believes. B believes X :then. A believes X
```

respectively. Now to establish "A believes X" it will suffice to establish

```
A believes. B believes X
```

This gap is usually closed by a hypothetical argument in which A reasons along the following lines:

If B believed what A believes about what C says (usually expressed by saying "if B were aware of the facts") then B would believe

```
C says X
```

hence B would believe X. So A believes that B is entitled to believe X, under circumstances which are in fact the case, therefore by our use of "belief"

```
A believes. B believes X
```

QED.

The fact that this "proof" nowhere uses B's honesty gives the game away: A has no legitimate basis to conclude anything about what B would actually believe that C says.

But surely what C says is a matter of fact, not of anybody else's belief? As an example to the contrary, consider a "secure" message delivery system, Ted, currently responsible for delivering a message M to Bob. If Ted finds himself under physical threat (eg power loss) then at least two different security policies are possible, depending upon whether disclosure or non-delivery is considered the greater threat:

(1) destroy the message M

(2) broadcast the message "Bob: M"

Since it may be important to conceal the fact of which policy is in operation, when power is cut, Ted may well broadcast the message "Bob: Aunt Agatha will arrive on the 5:15 from King's Cross." Bob knows that this message really means that Ted has succeeded in destroying all trace of the secret message M. The fact that Alice trusts Bob and knows that Bob trusts Ted is therefore not sufficient reason for Alice to go to the station expecting to meet Aunt Agatha. (Note that Bob has not betrayed Alice's trust, since Bob has not said anything!)

When we wrote

```
A believes. C says X
```

we blurred "C says X to/for A" with "C says X to/for B". What was needed to make the proof work was actually that

```
A believes. B says. C says X to/for B .to/for A
```

and that

```
A trusts B about. C says X to/for B
```

4 Conclusion.

The ability to localize trust is arguably the most important benefit of public-key cryptography. Performance optimization of distributed systems usually involves

the delegation of certain functions to "third-parties" that have their own policy and agenda. In order to allow an adequate analysis of the effect of these optimizations on the trust relationships of the system, it is important to remember that such an exercise consists not in establishing the properties actually possessed by the system, but rather in the analysis of the beliefs held by principals about each other and the basis upon which these beliefs are held. This analysis must consider intensional conditions of the principals (including modalities such as obligation and deceit) and not merely extensional properties of the system and the facts which make them true. Intensional conditions are not referentially transparent. In particular, Alice's analysis of Bob's beliefs must consider situations which Alice believes to be not merely false in actual fact, but impossible per se.

References

1. Burrows, M., Abadi, M., Needham, R.M., 1990, A Logic of Authentication, ACM Transactions on Computer Systems **8**(1) 18–36
2. Cheswick, W.R., Bellovin, S.M., 1994, Firewalls and Internet Security: Repelling the Wily Hacker, Addison Wesley, 0-201-63357-4
3. Cresswell, M.J., 1973, Logics and Languages, Methuen, 0-416-76950-0
4. Gallin, D., 1975, Intensional and Higher-Order Modal Logic: with Applications to Montague Semantics, North Holland, 0-7204-0360-X
5. Gong, L., Needham, R., Yahalom, R., 1990, Reasoning about Belief in Cryptographic Protocols, IEEE Computer Society Symposium on Research in Security and Privacy 1990, 234–248

This article was processed using the LaTeX macro package with LLNCS style