A Calculus for Distrust and Mistrust

Giuseppe Primiero(✉)

Department of Computer Science, Middlesex University, London, UK
G.Primiero@mdx.ac.uk

Abstract. Properties of trust are becoming widely studied in several applications within the computational domain. On the contrary, negative trust attribution is less well-defined and related issues are yet to be approached and resolved. We present a natural deduction calculus for trust protocols and its negative forms, distrust and mistrust. The calculus deals efficiently with forms of trust transitivity and negative trust multiplication and we briefly illustrate some possible applications.

1 Introduction

In various areas of the computational sciences, characterizations of trust are used to identify relevant, secure or preferred sources, channels and contents. For trust interpreted as a first order relation between agents, propagation needs to be considered [3,5,10,11]:

Example 1 (Trust Transitivity). If Alice trusts Bob and Bob trusts Carol; should Alice trust Carol?

This is undesirable in many security contexts. Solutions to this problem include decentralised trust [1], bounded-transitivity in authorization contexts [4], and a constraint by guarantors in [6]. In [18], trust is defined as a second-order property of first-order relations (e.g. of communication) between agents. This is applied in [17] to formulate SecureND, a proof-theoretic access control model with an explicit trust function over resources: agents do not trust other agents, but the information they receive from them. Informally, the trust function is defined as follows:

Definition 1 (Trust). *If Alice reads ϕ from Bob and ϕ is consistent with her profile, Alice trusts ϕ and can write it.*

SecureND resolves unintended transitive trust by requiring explicit localisation of trusted messages in the agents' profiles, similar to what suggested in [6].

Recently, research has started considering the different meanings of negative trust [9,13–15,20]. In the social sciences distrust is response to lack of information [7,8] and mistrust is former trust destroyed or healed [19]; the contextual account [13] present mistrust as misplaced trust, untrust as little trust and distrust as no trust. This approach abstracts from the reasons behind the attribution of these evaluations, in favour of a purely quantitative approach. Most of the

S.M. Habib et al. (Eds.): IFIPTM 2016, IFIP AICT 473, pp. 183–190, 2016.
DOI: 10.1007/978-3-319-41354-9_15

remaining contributions do not distinguish mistrust from distrust. Propagation for negative (first-order) trust is formulated as follows [12]:

Example 2 (Untrust Multiplication). If Alice does not trust Bob and Bob does not trust Carol; should Alice trust Carol?

In this paper, we introduce (un)SecureND, an extension of the calculus in [17] with rule-based definitions for negative trust over resources. Here and in the following we use the term *untrust* as neutral with respect to its derivatives *mistrust* for misplacement of trust, and *distrust* for betrayal. Our contribution distinguishes among these two terms, based on the intentional characterization offered in [16]. This calculus also resolves the problem of untrust multiplication. Consider the following modified example:

Example 3 (Intentional Untrust Multiplication). Alice does not trust ϕ from Bob: she believes he sends her *intentionally* false information. Bob does not trust $\neg\phi$ from Carol: he believes she sends him *intentionally* false information. Should Alice trust $\neg\phi$ from Carol?

The question is now better specified and we believe can be answered in the affirmative, given Carol's intention to deceive Bob, and Bob's intention to deceive Alice. The related epistemic action of *distrust* has the following intuitive semantics:

Definition 2 (Distrust). *If Alice reads ϕ from Bob and ϕ is inconsistent with Alice's profile, Alice distrust ϕ and writes $\neg\phi$.*

A distinct case for trust misplacement can be formulated as follows:

Example 4 (Unintentional Untrust Multiplication). Alice reads ϕ from Bob, false in view of her current information: she believes she has *unintentionally* held false information $\neg\phi$. Bob has received ϕ from Carol, who can confirm it to Alice. Should Alice trust ϕ from Carol?

The intuitive semantic meaning of this form of negated trust is as follows:

Definition 3 (Mistrust). *If Alice reads ϕ from Bob, ϕ is inconsistent with Alice's profile and Alice wants to maintain consistency, then she either mistrusts $\neg\phi$; else she refuses ϕ.*

To accept or reject such contradicting information might depend on the number and role of other agents available for confirmation.

The rest of the paper is structured as follows. In Sect. 2 we introduce the natural deduction calculus (un)SecureND: it defines protocols by which agents trust, mistrust or distrust information based on an intentional interpretation of the truth of data transmission; we also briefly cover its meta-theoretical properties. In Sect. 3 we illustrate the restriction to untrust multiplication allowed by this calculus and informally present a possible application to software management, extending the work in [2]. In Sect. 4 we survey further research directions.

2 (un)SecureND

(un)SecureND is a natural deduction calculus defining trust, mistrust and distrust protocols. It formalizes a derivability relation on formulas from sets of assumptions (contexts) as accessibility on resources issued by agents.

Definition 4 (Syntax of (un)SecureND).

$$\begin{aligned}
&\mathcal{S}^{\sim} := \{A \leq B \leq \dots\} \\
&BF^S := a^S \mid \phi_1^S \rightarrow \phi_2^S \mid \phi_1^S \wedge \phi_2^S \mid \phi_1^S \vee \phi_2^S \mid \bot \\
&mode := Read(BF^S) \mid Write(BF^S) \mid Trust(BF^S) \\
&RES^S := BF^S \mid mode \mid \neg RES^S \\
&\Gamma^S := \{\phi_1^S, \dots, \phi_n^S\};
\end{aligned}$$

$\mathcal{S}$ is a set of subjects, with a partial order relation $\leq$ over $\mathcal{S} \times \mathcal{S}$: intuitively, $S \leq S'$ means that subject S has higher security privileges than S'. The partial order allows for branching in the hierarchy, so that e.g. $A < B < C$ and $A < B < D$, but C, D are not comparable. BF^S is a set of boolean formulae inductively defined by logical connectives and including $\bot$ for the false. *mode* is a variable for reading, writing and trusting formulae. Formulae and functions are closed under negation. $\vdash \phi^A$ indicates a validly derivable resource ϕ issued by agent A. Context Γ^A formalises a set of formulae describing the profile for agent A, under which some other resource can be accessed. A context can be extended by a formula issued by the same agent, denoted by Γ^A, ϕ^A; or it can be extended by resources from a different agent, denoted by $\Gamma^A; \phi^B$ and $\Gamma^A; \Gamma^B$.

Definition 5. *An* (un)SecureND*-formula* $\Gamma^A \vdash RES^B$ *says that under the profile for user* A*, some resource from user* B *is validly accessed, given* $A \sim B$*.*

The calculus is based on two sets of rules. The access order to be applied to these rules can be specified dependently on the application: for example, to implement a downwards-only access protocol, the rules will hold only if $A < B$. The operational rules to introduce and eliminate connectives on resources across agents are given in Fig. 1. The rule *Atom* establishes derivability of formulae included in well-formed contexts and preserved under extension. We use the abbreviation wf for a profile that preserves consistency construable by induction from the empty profile. $\wedge$-I says that if ϕ_1^A is derivable from profile Γ^A and ϕ_2^B is derivable from profile Γ^B, then their conjunction is derivable from the joint profiles. By the elimination, each composing resource is derivable from the combined profiles. $\vee$-I says that if a joint profile for users A, B can access a formula ϕ_i^I, then it can access the disjunction with any other formula. By the elimination, each resource ψ^I derivable from each component ϕ_i^I can also be obtained by the extended profile. $\rightarrow$-Introduction establishes the validity of the Deduction Theorem; its elimination implements Modus Ponens. Negation is defined (in the standard constructive way) by implication to the false.

In Fig. 2 we present the access rules allowing a user's profile to act on resources available from another user. $\neg$-distribution implements a form of negation-completeness: if a profile cannot access a resource from another agent,

$$\frac{\Gamma^A \vdash wf}{\Gamma^A; \Gamma^B \vdash b} \text{ Atom, for any } b \in \Gamma^B$$

$$\frac{\Gamma^A \vdash \phi_1^A \qquad \Gamma^B \vdash \phi_2^B}{\Gamma^A; \Gamma^B \vdash \phi_1^A \wedge \phi_2^B} \wedge\text{-I} \qquad \frac{\Gamma^A; \Gamma^B \vdash \phi_1^A \wedge \phi_2^B}{\Gamma^A; \Gamma^B \vdash \phi_i^I} \wedge\text{-E}$$

$$\frac{\Gamma^A; \Gamma^B \vdash \phi_i^I}{\Gamma^A; \Gamma^B \vdash \phi_1^A \vee \phi_2^B} \vee\text{-I} \qquad \frac{\Gamma^A; \Gamma^B \vdash \phi_1^A \vee \phi_2^B \qquad \phi_i^I \vdash \psi^I}{\Gamma^A; \Gamma^B \vdash \psi^I} \vee\text{-E}$$

with $I \in \{A, B\}, i \in \{1, 2\}$ in the above rules.

$$\frac{\Gamma^A; \phi_1^B \vdash \phi_2^B}{\Gamma^A \vdash \phi_1^B \to \phi_2^B} \to\text{-I} \qquad \frac{\Gamma^A \vdash \phi_1^B \to \phi_2^B \qquad \Gamma^A \vdash \phi_1^B}{\Gamma^A; \phi_1^B \vdash \phi_2^B} \to\text{-E}$$

$$\frac{\Gamma^A \vdash RES^A \to \bot}{\Gamma^A \vdash \neg RES^A} \; bot$$

Fig. 1. The system `(un)SecureND`: operational rules

then it can access its negation (although strong, this rule is essential to preserve consistency). *read* says that from any well-formed profile A, formulae from a profile B can be read (this will hold according to the required constraint on the order relation among agents). *trust* says that if a resource can be read and it preserves consistency when added to the reading profile, then it can be trusted. *write* says that a readable and trustable resource can be written. By DTrust, agent A distrusts a resource ϕ^B if it induces contradiction when read from Γ^A. Its elimination uses $\to$-introduction to induce *write* from the receiver profile of any resource that follows distrusting operations. This trivially allows $Write(\neg\phi^B)$ when $\neg Trust(\phi^B)$ holds. By MTrust, agent A mistrusts resource $\phi^A \in \Gamma^A$ if it contradicts some received ψ^B; then $Cn(\phi^A)$ is removed to accommodate ψ^B in Γ^A. Its elimination depends on a checking operation. By MTrust-E1, if at least one C agent higher in the order than the sender B verifies the information ϕ^A originally held by the receiver A, ψ^B is rejected; if the receiving agent is the only one higher in the order relation with respect to the sender, the mistrust operation reduces to a distrust one; for $C < B < A$, the receiver A looks for all agents with higher reputation and/or privileges than sender B in order to check for the content of the message ψ. By MTrust-E2, if for every agent C higher than the sender B verifies the received contradictory information ψ^B, the receiver A removes ϕ^A from her profile and trusts the new information.

2.1 Metatheory

The following standard meta-theoretical properties hold for (`un`)`SecureND` under trust, all proofs are formulated by structural inductions on the derivation of the second assumption (omitted for brevity).

$$\frac{\Gamma^A \vdash \neg mode(\phi^B)}{\Gamma^A \vdash mode(\neg\phi^B)}\ \neg\text{-distribution} \qquad \frac{\Gamma^A \vdash wf}{\Gamma^A \vdash Read(\phi^B)}\ read$$

$$\frac{\Gamma^A \vdash Read(\phi^B) \qquad \Gamma^A; \phi^B \vdash wf}{\Gamma^A \vdash Trust(\phi^B)}\ trust$$

$$\frac{\Gamma^A \vdash Read(\phi^B) \qquad \Gamma^A \vdash Trust(\phi^B)}{\Gamma^A \vdash Write(\phi^B)}\ write$$

$$\frac{\Gamma^A \vdash wf \qquad \Gamma^A \vdash Read(\phi^B) \to \bot}{\Gamma^A \vdash \neg Trust(\phi^B)}\ \text{DTrust-Intro}$$

$$\frac{\Gamma^A \vdash \neg Trust(\phi^B) \qquad \Gamma^A \vdash \neg Trust(\phi^B) \to \psi^A}{\Gamma^A \vdash Write(\psi^A)}\ \text{DTrust-Elim}$$

$$\frac{\Gamma^A \vdash Read(\psi^B) \to \bot \qquad \Gamma \setminus \{\phi^A\} \vdash wf, \forall \phi^A \vdash Read(\psi^B) \to \bot}{\Gamma \setminus \{\phi^A\}; \psi^B \vdash \neg Trust(\phi^A)}\ \text{MTrust-Intro}$$

$$\frac{\Gamma \setminus \{\phi^A\}; \psi^B \vdash \neg Trust(\phi^A) \qquad \Delta^C \vdash Read(\psi^B) \to \bot}{\Gamma^A; \Delta^C \vdash Trust(\phi^A)}\ \text{MTrust-E1, for } C < B$$

$$\frac{\Gamma \setminus \{\phi^A\}; \psi^B \vdash \neg Trust(\phi^A) \qquad \Delta^C; \psi^B \vdash wf}{\Gamma \setminus \{\phi\}^A; \Delta^C \vdash Trust(\psi^B)}\ \text{MTrust-E2, } \forall C < B$$

Fig. 2. The system `(un)SecureND`: access rules

Theorem 1 (Weakening $A \sim B$). *If $\Gamma^A \vdash Write(\phi^A)$ and $\Gamma^A \vdash Trust(\psi^B)$, then $\Gamma^A; \psi^B \vdash Write(\phi^A)$.*

Theorem 2 (Contraction $A \sim B$). *If $\Gamma^A, \phi^A; \phi^B \vdash Write(\psi^A)$, then $\Gamma^A, \phi^A \vdash Write(\psi^A)$.*

Theorem 3 (Exchange $A \sim B$). *If $\Gamma^A, \phi^A; \psi^B \vdash \rho^A$, then $\Gamma^A; \psi^B; \phi^A \vdash \rho^A$.*

The general form of the cut rule is as follows:

$$\frac{\Gamma^A \vdash \phi^B \qquad \Delta^B, \phi^B \vdash \psi^B}{\Gamma^A; \Delta^B \vdash \psi^B}\ \text{Cut}$$

With $A < B$, it amounts to a cut downwards the order relation; with $B < A$ to one upwards: which one is allowed depends again on the application. If $\phi^B \equiv \neg Trust(\phi^B)$ and $A < B$, then the first premise is the result of a DTrust rule, the second premise result from a MTrust rule, and the cut rule eliminates both;

if $\phi^B \equiv \neg Trust(\phi^A)$, the first premise is obtained by a MTrust rule, the second from a DTrust rule. In all these cases the conclusion of Cut will be an instance of a Weakening rule. If $\psi^B \equiv \neg Trust(\psi^B)$, then all cases reduce to instances of Weakening on conclusions of a MTrust rule. Then untrust relations safely extend the following from [17]:

Theorem 4 (Cut-Elimination Theorem). *Any* (un)`SecureND` *derivation with an instance of a Cut-rule can be transformed into another derivation with the same end sequent iff appropriate trust-access is granted on any upward domination relation among agents.*

3 Examples and Applications

In [17] trust transitivity from Example 1 is resolved by explicitly guaranteeing consistency on every access to resources within the current profile. If Alice trusts ϕ from Bob, and Bob trusts ψ from Carol, Alice also trusts (and eventually writes) information ψ from Carol iff extending her profile Γ^A with information ϕ^B and ψ^C is explicit and preserves consistency.

In (un)`SecureND`, untrust multiplication from Example 2 is restricted to *distrust*, i.e. all agents involved are actively trying to deceive their trustor:

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma^B \vdash wf \qquad \Gamma^B \vdash Read(\neg\phi^C) \to \bot}{\Gamma^B \vdash \neg Trust(\neg\phi^C)} \qquad \Gamma^B \vdash Write(\phi^B)}{\Delta^A \vdash \neg Trust(\phi^B)} \qquad \Delta^A \vdash Read(\phi^B) \to \bot}{\Delta^A \vdash Trust(\neg\phi^C)} \qquad \Delta^A; \neg\phi^C \vdash wf}{\Delta^A \vdash Write(\neg\phi^C)}$$

If Alice believes Bob is trying to deceive her with ϕ, and Bob believes Carol is trying to deceive him with $\neg\phi$, then Alice can trust $\neg\phi$ from Carol.

`SecureND` has been applied to the Minimally Trusted Install Problem in [2]: determine the way to install a new package p in a system such that the minimal amount of transitively trusted dependencies for p is satisfied. In `(un)SecureND` we can resolve the negative counterpart of this problem. We offer here only an informal explanation and leave a full formalization and the extension of the Coq protocol from [2] to further research. Consider an installation profile Γ^A, and a software package ψ available from repository B for installation. DTrust-Intro can be applied to return all packages that have unresolved conflicts in Γ^A and as such cannot be installed, including ψ^B. DTrust-Elim returns all packages that can be installed under the current conflict with ψ^B. MTrust-Intro returns all packages already installed in Γ^A that need to be removed for Γ^A to install ψ^B safely. MTrust-E1 returns all external packages that can be installed in Γ^A preserving the *current* installation and hence the conflict with ψ^B. MTrust-E2 returns all packages that can be safely installed in Γ^A preserving the installation of ψ^B.

4 Conclusions

(Un)trust relations reveal relevant problems for privacy and security. Attackers can exploit negative trust to induce unconstrained positive information; intentional transmission of true data can be conceived as a strategy to win the trustor's confidence for future attacks, with trustworthiness evaluation based on records of high rate of false alarms (or low records of true alarms). Untrust multiplication can generate unintended accesses and operations. An evaluation based on intentionality criteria can offer a sensibly better solution in many cases if combined with a quantitative and computationally feasible approach. We have presented a calculus for access control protocols with negative trust, modelled formally as functions on resources issued by agents. This language qualifies trust transitivity under consistency constraints and limits untrust multiplication to intentional cases of false data transmission. It also allows revision of false content held within an agent's profile in the form of mistrust. Next stages of this research will focus on defining structural weakenings of the calculus and the development of applications.

References

1. Abdul-Rahman, A., Hailes, S.: A distributed trust model. In: Haigh, T., Blakley, B., Zurko, M.E., Meodaws, C. (eds.), Proceedings of the 1997 Workshop on New Security Paradigms, Langdale, Cumbria, United Kingdom, September 23–26, 1997, pp. 48–60. ACM (1997)
2. Boender, J., Primiero, G., Raimondi, F.: Minimizing transitive trust threats in software management systems. In: Ghorbani, A.A., Torra, V., Hisil, H., Miri, A., Koltuksuz, A., Zhang, J., Sensoy, M., García-Alfaro, J., Zincir, I. (eds.) 13th Annual Conference on Privacy, Security and Trust, PST 2015, Izmir, Turkey, July 21–23, 2015, pp. 191–198. IEEE (2015)
3. Chakraborty, P.S., Karform, S.: Algorithms, designing trust propagation based on simple multiplicative strategy for social networks. Procedia Technol. **6**, 534–539 (2012). 2nd International Conference on Communication, Computing & Security [ICCCS-2012]
4. Chapin, P.C., Skalka, C., Wang, X.S.: Authorization in trust management: features and foundations. ACM Comput. Surv. **40**(3), 1–48 (2008)
5. Christianson, B., Harbison, W.S.: Why isn't trust transitive? In: Crispo, B. (ed.) Security Protocols 1996. LNCS, vol. 1189, pp. 171–176. Springer, Heidelberg (1997)
6. Christianson, B.: Trust*: using local guarantees to extend the reach of trust. In: Christianson, B., Malcolm, J.A., Matyáš, V., Roe, M. (eds.) Security Protocols 2009. LNCS, vol. 7028, pp. 179–188. Springer, Heidelberg (2013)
7. Cvetkovich, G.: The attribution of social trust. In: Cvetkovih, G., Lofstedt, R. (eds.) Social Trust and the Management of Risk, pp. 53–61. Earthscan, London (1999)
8. Cvetkovich, G., Lofstedt, R.E.: Social trust and culture in risk management. In: Cvetkovih, G., Lofstedt, R. (eds.) Social Trust and the Management of Risk, pp. 9–21. Earthscan, London (1999)

9. Guha, R.V., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: Proceedings of the 13th International Conference on World Wide Web, WWW 2004, New York, NY, USA, May 17–20, 2004, pp. 403–412 (2004)
10. Jamali, M., Ester, M.: A Matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys 2010, pp. 135–142. ACM, New York (2010)
11. Jøsang, A., Marsh, S., Pope, S.: Exploring different types of trust propagation. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 179–192. Springer, Heidelberg (2006)
12. Jøsang, A., Pope, S.: Semantic constraints for trust transitivity. In: Hartmann, S., Stumptner, M. (eds.), APCCM, vol. 43 of CRPIT, pp. 59–68. Australian Computer Society (2005)
13. Marsh, S., Dibben, M.R.: Trust, untrust, distrust and mistrust – an exploration of the Dark(er) side. In: Herrmann, P., Issarny, V., Shiu, S.C.K. (eds.) iTrust 2005. LNCS, vol. 3477, pp. 17–33. Springer, Heidelberg (2005)
14. McKnight, D.H., Chervany, N.L.: Trust and distrust definitions: one bite at a time. In: Falcone, R., Singh, M., Tan, Y.-H. (eds.) AA-WS 2000. LNCS (LNAI), vol. 2246, pp. 27–54. Springer, Heidelberg (2001)
15. McKnight, D.H., Kacmar, C., Choudhury, V.: Whoops..did i use the wrong concept to predict e-commerce trust? Modeling the risk-related effects of trust versus distrust concepts. In: 36th Hawaii International Conference on System Sciences (HICSS-36 2003), CD-ROM / Abstracts Proceedings, January 6–9, 2003, Big Island, HI, USA, p. 182 (2003)
16. Primiero, G., Kosolosky, L.: The semantics of untrustworthiness. Topoi **35**(1), 253–266 (2013)
17. Primiero, G., Raimondi, F.: A typed natural deduction calculus to reason about secure trust. In: Miri, A., Hengartner, U., Huang, Audun Jøsang, N.-F., García-Alfaro, J. (eds.), 2014 Twelfth Annual International Conference on Privacy, Security and Trust, Toronto, ON, Canada, July 23–24, 2014, pp. 379–382. IEEE (2014)
18. Primiero, G., Taddeo, M.: A modal type theory for formalizing trusted communications. J. Appl. Logic **10**, 92–114 (2012)
19. Sztompka, P.: Trust: A Sociological Theory. Cambridge University Press, Cambridge (1999)
20. Ziegler, C.-N., Lausen, G.: Propagation models for trust and distrust in social networks. Inf. Syst. Front. **7**(4–5), 337–358 (2005)