A typed natural deduction calculus to reason about secure trust

Giuseppe Primiero
Department of Computer Science
Middlesex University, UK
Email: G.Primiero@mdx.ac.uk

Franco Raimondi
Department of Computer Science
Middlesex University, UK
Email: F.Raimondi@mdx.ac.uk

***Abstract*—System integrity can be put at risk by unintentional transitivity of resource access. We present a natural deduction calculus for an access control model with an explicit trust function on resources. Its inference relation is designed to limit unintentionally transitive access from untrusted parties. We also offer results for ordered cut and normalization related to security and hint at a prototype implementation.**

I. Introduction

An important threat to system integrity and security through access control is represented by promiscuous or unintentionally transitive trust (see [9], [8]): *"Bob trusts Carol. Alice trusts Bob, but she has no way of knowing that he trusts Carol. Then Alice has no control on the risks to which she is exposed when trusting Bob."* In this scenario, access privileges of third parties might need restriction. One way to limit such inconvenient system behaviour is to require that resource access depends from explicit trust among all subjects: *"Bob trusts resources from Carol, and similarly Alice from Bob. Alice can access and use Carol's resources if she trust them explicitly"*. This policy requires explicit trust declaration of Carol from Alice and implementation of this localisation principle motivates our contribution (see [10]).

Such explicit trust treatment has not yet been explored in the context of well-known access control models. In the Bell-LaPadula Model (BLPM, [3]), a subject can read only resources whose access group is dominated by the subject's access group ("no read up") and can write only resources whose access group dominates the subject's access group ("no write down"); a trusted subject is allowed to violate the writing constraint above, if it is not against security by design. Trust is simply intended as a property of agents, while security is a property of the system defined independently of the former. In the Role Based Access Control security model (RBAC, [11]), security is enhanced by taking into account role relations that subjects have in a system. Subjects on lower integrity levels are not permitted to write resources on higher integrity levels ("no write up" rule); and subjects on higher integrity levels cannot be corrupted by accessing resources on lower integrity levels ("no read down"). Trustworthiness of resources corresponds to prevention of unauthorized change. Trust-aware RBAC models have been recently explored, in which trustworthiness is either defined by temporal-spatial constraints (e.g. [4], [7]), level-constraints ([6]) or by explicitly requiring role related assessments based on the behavior history of the user, see e.g. [15] for a subjective logic model. A constrained transitive trust model for the latter logic is also offered in [13]. In authentication logic, starting with [1], beliefs are explicitly used: its `says` modality for principals can be interpreted to include a trust relation, as fully explored in [12], with application to distributed setting, [2]. Transitivity of trust is also analysed in the context of cryptographic applications, see e.g. [14]. We offer a calculus `SecureND` for secure transitive trust on access resources in the unexplored setting of typed natural deduction. An advantage of this approach is that it displays a substructural-like inference relation restricting transitivity by limiting Weakening. Moreover, all meta-properties of the system have a direct translation into security aspects. Our system includes the following access modes on resources: `r-attribute` (an agent performs some reading), `w-attribute` (an agent performs some writing) and `t-attribute` (an agent trusts some resource issued by some agent). Trust is therefore treated as a function ranging over resources and not as a relation between subjects:

Definition 1 (Trust)**.** *Assume that A's protocol grants access to B's resource ϕ. Trust characterizes access to ϕ iff when reading ϕ, A is able to make ϕ part of its own protocol. If A trusts ϕ from B, then A is able to write ϕ.*

This implies a shift from the reading *'Admin trusts Bob to do so and so on resource ϕ'*, to *'Under Admin's protocol, access to resource ϕ from Bob is trustable'*. This informal notion is matched formally in the $trust$ rule of the calculus, with the following meaning: if under A's protocol a resource ϕ from B can be read, and ϕ can validly extend A's protocol, then resource ϕ from B is trusted under A's protocol. Valid protocol extension is interpreted by resource typability. The resulting policy for access attributes is a basic "read-down/write-down": reading privileges transfer monotonically downwards a domination relation between subjects; writing privileges are dependent on the possibility of making messages part of the agent's accessible resources; security is expressed in the form of reading-writing constraints under trust relations; a "write-up" policy is implementable by defining trust on an upward domination relation.

II. The Calculus SecureND

SecureND is a typed natural deduction calculus designed for secure operations on resources issued by subjects with different privileges. Reading and writing access privileges are extended with a bridging *trust* access mode. Its use requires a form of Weakening (see Proposition 1) and it satisfies Transitivity (blocking it when not allowed).

Definition 2 (Syntax)**.** *The syntax of* SecureND *is defined by the following alphabet:*

$$\mathcal{S} := \{A, B, \dots\}$$
$$BF^S := a^S \mid \phi_1^S \to \phi_2^S \mid \phi_1^S \wedge \phi_2^S \mid \phi_1^S \vee \phi_2^S$$
$$mode := Read(BF^S) \mid Write(BF^S) \mid Trust(BF^S)$$
$$RES^S := BF^S \mid mode$$
$$\Gamma^S := \{\phi_1^S, \dots, \phi_n^S\};$$

with $\mathcal{S}$ a set of subjects, BF^S boolean formulae inductively defined by connectives, $mode$ for access functions over formulae, RES^S including both contents and access modes, and Γ^S the protocol for S under which access is operated.

Definition 3. *A* SecureND*-sequent $\Gamma^A \vdash RES^B$ says that under the protocol for subject A, some resource for subject B is valid or accessed.*

By protocol construction in Definition 2, it holds that $\Gamma^A; \cdot \vdash wf$ iff $\Gamma^A \neq \emptyset$ and $\forall \psi^S \in \Gamma^A$, $S = A$, that is to say the protocol is not empty and every formula contained in it is valid for the subject typing the protocol. Protocol extension is explained as follows: (Γ, ϕ) corresponds to subject's protocol extension Γ^A, ϕ^A; the extension by distinct contexts $(\Gamma; \phi)$, with $\phi \in \Delta$ corresponds to protocols addition by different subjects $\Gamma^A; \phi^B$. Validity of context extension requires therefore subjects' privileges classification. A partial order relation $\leq$ over $\mathcal{S} \times \mathcal{S}$ models the dominance relation between subjects. Intuitively, $S \leq S'$ means that subject S has higher security privileges than subject S'. Valid privilege transfers for access control are summarized as follows:

- $\Gamma^S \vdash Read(\phi^S)$ and $Write(\phi^S)$ hold;
- $\Gamma^S \vdash Read(\phi^{S' \geq S})$ holds;
- $\Gamma^S \vdash Write(\phi^{S' \geq S})$ holds under trust;
- $\neg\forall(S', S), \Gamma^S \vdash Trust(\phi^{S'})$.

While the ordering of the domination relation is transitive (if $S < S'$ and $S' < S''$, then $S < S''$), if S reads from S' and S' writes content from S'', then S reads from S'' iff S' trusts content from S''. In particular, it will be not the case that if $S < S' < S''$ and S reads from S', and S' writes content from S'', then S reads from S''. Reading and writing is thus trivial within the same protocol. When a subject reads content from another subjects, we do not extend such privileges directly to writing. The rules system SecureND is introduced in Figure 1 and assumes that $A \leq B$ holds, and $I \in \{A, B\}; i \in \{1, 2\}$. $Atom$ says that any atomic content is accessible from a well-formed protocol higher in the dominance relation. $\wedge$ constructs and destructs contents accessible from different protocols. $\vee$ constructs and destructs contents accessible from combined protocols. $\to$ establishes the validity of the Deduction Theorem for contents downward accessible, elimination implements Modus Ponens. By $read$, given a well-formed protocol Γ^A, content ϕ^B is readable ("read-down"). By $trust$, assuming a read-down and that the content is made part of the subject's protocol, means that the content can be trusted. By $write$ a content ϕ^B readable and trustable under Γ^A, can be written ("write-down" under trust).

III. Security by Normalization

The operation of context extension by differently typed resources needs to preserve well-formedness on protocol construction:

Proposition 1 (Well-formed Import)**.** *If $\Gamma^A \vdash Read(\phi^B)$ and $\{\Gamma^A, \phi^B\} \in RES^A$, then $\Gamma^A; \phi^B \vdash wf$.*

Security threats might occur if A does not have the right privileges over contents from B. We show how secure access requires trusted import operations, by proving them equivalent to access performed under a unique protocol. Such good behaviour corresponds in a natural deduction calculus to prove a form of cut-elimination theorem: we first interpret the import as cut-rules in a upward and downward form (see Figure 1); then we show that any step in an access operation in SecureND containing such a rule can be eliminated without loss of information. For the upward version this requires explicit trust. We shall refer to a *redex* for a derivation before the cut is eliminated; a *contractum* for the derivation after the cut is eliminated; and *descendent* for the sub-derivation in the contractum deriving after the eliminated cut. Validity of $\downarrow Cut$ depends on implementing $import$ in the form of a "write-down" rule: if $\Gamma^A \vdash Write(\Gamma^B)$ and $\{\Gamma^A; \Gamma^B\} \in RES^A$, then $\Gamma^A; \Gamma^B \vdash wf$.

Theorem 1 (↓Cut-Elimination Theorem)**.** *Any* SecureND *access operation with an import by an occurrence c of the $\downarrow Cut$ rule can be transformed into another operation with the same final access without c using only (downward) well-formed import.*

Proof. By induction on the derivation D which is the redex of the cut-elimination. Assuming c is the only ↓ Cut rule and it is the last inference rule of the redex, the derivation D' which is the contractum of the cut-elimination contains a descendent of the cut obtained by a downward import according to the domination relation; because the formula obtained by the cut is, by hypothesis, derivable from the weaker protocol, it will also be derivable from the weaker and the stronger protocol together. When c is not the last inference rule of the redex, then the descendent of the cut will admit all downwards imports preserving the one occurring in the cut; those imports will occur also in the contractum of the cut rule and can be traced back up to the one formulation of the downward import that occurs in the cut rule. □

Validity of $\uparrow Cut1$ requires $import$ as a "write-up" rule on individual resources: if $\Gamma^B \vdash Write(\phi^A)$ and $\{\Gamma^B; \phi^A\} \in RES^B$, then $\Gamma^B; \phi^A \vdash wf$. Validity of $\uparrow Cut2$ requires

$$\frac{\Gamma^A;\cdot\vdash wf}{\Gamma^A;\Gamma^B\vdash b}\ \text{Atom, for any } b\in\Gamma^B \qquad \frac{\Gamma^A\vdash\phi_1^A \qquad \Gamma^B\vdash\phi_2^B}{\Gamma^A;\Gamma^B\vdash\phi_1^A\wedge\phi_2^B}\ \wedge\text{-I} \qquad \frac{\Gamma^A;\Gamma^B\vdash\phi_1^A\wedge\phi_2^B}{\Gamma^A;\Gamma^B\vdash\phi_i^I}\ \wedge\text{-E}$$

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

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

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

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

$$\frac{\Gamma^A\vdash\phi^B \qquad \Gamma^B,\phi^B\vdash\psi^B}{\Gamma^A;\Gamma^B\vdash\psi^B}\downarrow Cut \qquad \frac{\Gamma^A\vdash\phi^A \qquad \Gamma^B;\phi^A\vdash\psi^B}{\Gamma^A;\Gamma^B\vdash\psi^B}\uparrow Cut1 \qquad \frac{\Gamma^B\vdash\phi^B \qquad \Gamma^A;\phi^B\vdash\psi^A}{\Gamma^B;\Gamma^A\vdash\psi^A}\uparrow Cut2$$

Fig. 1. The system SecureND

import as a "write-up" rule on full protocols: if $\Gamma^B \vdash Write(\Gamma^A)$ and $\{\Gamma^B;\Gamma^A\} \in RES^B$, then $\Gamma^B;\Gamma^A \vdash wf$. In general, under a "no write-up" policy, neither of the upward cuts will be considered valid for $A \leq B$. "Write-up" rights are permissible if and only explicit trust on typed resources is given.

Theorem 2 (↑Cut-Elimination Theorem)**.** *Any* SecureND *access operation with an import by an occurrence c of the $\uparrow Cut1/2$ rule can be transformed into another operation with the same final access without c using only (upward) well-formed import iff appropriate $trust$-access is granted for the involved subjects.*

Proof. By induction on either the premise of the derivation D which represents the redex of the cut elimination (for $\uparrow Cut1$) or on the conclusion of the derivation D' which represents the contractum of the cut elimination (for $\uparrow Cut2$). Assuming this is the only cut and the last rule of the redex or contractum derivation. For both cases: the condition expressed by the second premise of the relevant import is the result of applying a $trust$ rule, provided that in the first case the cut formula typed for A is required to extend the protocol for B and in the second case the protocol for B is required to be extended by the protocol for A. Hence, for both extensions a requirement that either the formula or the protocol are trusted by the subject lower in the domination relation holds; this condition being satisfied, the reduction of the cut elimination can be performed. □

Normalization for SecureND derivations reduces to a security constraint depending on trusted relations only:

Theorem 3 (Normalized secure derivations)**.** *All access operations of* SecureND *with read-writing access attributes are secure iff they implement trust attributes when involving subjects in an upward dominating relation.*

Proof. Normalization for a redex containing a $Read$ access attribute is obtained in the form of a detour elimination through a $Write$ access attribute:

$$\frac{\dfrac{\Gamma^A;\cdot\vdash wf}{\Gamma^A\vdash Read(\phi^B)}\ read \qquad \Gamma^A\vdash Trust(\phi^B)}{\Gamma^A\vdash Write(\phi^B)}\ write$$

$$\rightsquigarrow$$

$$\frac{\Gamma^A;\cdot\vdash wf}{\Gamma^A\vdash Write(\phi^B)}\ Write$$

The normalization holds directly for all $\phi^B \in \Gamma^B$ when $A < B$, i.e. under a read-down policy; $trust$ is essential for upward domination relations $B < A$, i.e. to implement a write-up policy.

□

IV. Structural results for Integrity

Theorem 4 (Weakening)**.** *Accessibility is preserved when protocols are extended by trusted resources:*

1) *If $\Gamma^A \vdash \phi^A$ and $\Gamma^A \vdash Read(\phi^B), \forall\phi \in \Gamma^B$, then $\Gamma^A;\Gamma^B \vdash \phi^A$.*
2) *If $\Gamma^B \vdash \phi^B$ and $\Gamma^B \vdash Trust(\phi^A), \forall\phi \in \Gamma^A$, then $\Gamma^A;\Gamma^B \vdash \phi^B$.*

Proof. By structural induction on the derivation tree of the second premise: for $A < B$, in 1. the import $\Gamma^A;\Gamma^B \in RES^A$ is satisfied by simple read attributes on B; in 2. the import $\Gamma^A;\Gamma^B \in RES^B$ requires trust attributes. □

Theorem 5 (Contraction)**.** *Accessibility within the same protocol or for $A < B$ is preserved when protocols present multiple occurrences of the same trusted resources:*

1) *If $\Gamma^A, \phi^A, \phi^A \vdash \psi^A$, then $\Gamma^A, \phi^A \vdash \psi^A$.*
2) *If $\Gamma^A, \phi^A; \phi^B \vdash \psi^A$, then $\Gamma^A, \phi^A \vdash \psi^A$.*

Proof. For 1., by the properties of SecureND. For 2., by structural induction on the derivation tree by using the domination relation and the resulting read-write attributes by using $trust$

between A, B to allow the import in A and then reduction to point 1.. □

Theorem 6 (Exchange)**.** *Accessibility within the same protocol or for $A < B$ is preserved when access to resources is inverted:*

1) *If $\Gamma^A, \phi^A, \psi^A \vdash \rho^A$, then $\Gamma^A, \psi^A, \phi^A \vdash \rho^A$.*
2) *If $\Gamma^A, \phi^A; \psi^B \vdash \rho^A$, and $\Gamma^A \vdash Trust(\psi^B)$, then $\Gamma^A; \psi^B; \phi^A \vdash \rho^A$.*

Proof. For 1., by the properties of SecureND. For 2., by structural induction on the derivation tree by using the domination relation and the resulting read-write accesses by using *trust* between A, B to allow the import in A and then reduction to point 1.. □

V. An example for non-promiscuous trust

We now show SecureND at work in the example from Section I. Alice can access and read from Bob; because Bob's contents are trusted, Alice can write them.

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

Bob can access and read from Carol; because Carol's contents are trusted, Bob can write them.

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

Joining the two access operations, one derives that under Alice's protocol, contents from Bob allow writing Carol's contents.

$$\frac{\Gamma^A \vdash Write(\phi^B) \qquad \phi^B \vdash Write(\psi^C)}{\Gamma^A; \cdot \vdash \phi^B \to Write(\psi^C)}\ \to\text{-I}$$

The *trust* function is needed to allow the import Γ^A, ϕ^B and writing of ψ^C under B's protocol $\{\phi^B, \psi^C\}$. Nonetheless, no transfer of privileges is yet made from B to A as far as content from C is concerned. This means that there is no derived SecureND sequent telling us that Alice is exposed to contents from Carol. Acceptability of ψ within A's protocol is expressed as an additional requirement under the *import* $\{\Gamma^A; \phi^B; \phi^C\} \in RES^A$. As a result of explicit trust, under A's protocol ψ^C is written and read. This means that Alice will need to have explicit information about whom is Bob trusting.

$$\frac{\Gamma^A; \cdot \vdash \phi^B \to Write(\psi^C) \qquad \Gamma^A; \cdot \vdash \phi^B \to Trust(\psi^C)}{\Gamma^A; \phi^B \vdash Write(\psi^C)}\ \to\text{-E}$$

VI. Conclusions and applications

SecureND offers a read-down only policy access, extended to a secure write-up under explicit trust attributes. This extension is allowed by an *import* principle with localized, subject's typed resources. In terms of security, this means a more strict and explicit control on subjects involved in access operations, guaranteeing access validity only when trust is explicitly expressed. An advantage of the rule-based structure of SecureND is the implementation for the theorem prover Coq, [5]. We have designed a library to check validity of a protocol that mimics the relation between trusted repositories in a software installation process. The full code is freely available at http://www.rmnd.net/MT/SecureND.v. The interpretation of SecureND in this context allows the control over new packages installation such that the minimal amount of transitively trusted dependencies from new repositories needs to be satisfied. Future developments will focus on the extension to the negative fragment of the language, to mimic behaviour in the presence of untrustworthy resources and, accordingly for the implementation, to identify all compatible installation profiles in view of package uninstall.

References

[1] Martín Abadi, Michael Burrows, Butler Lampson, and Gordon Plotkin. A calculus for access control in distributed systems. *ACM Trans. Program. Lang. Syst.*, 15(4):706–734, September 1993.

[2] Steve Barker and Valerio Genovese. Socially constructed trust for distributed authorization. In Vijay Atluri and Claudia Díaz, editors, *ESORICS*, volume 6879 of *Lecture Notes in Computer Science*, pages 262–277. Springer, 2011.

[3] D. E. Bell and L. J. LaPadula. Secure computer systems: Mathematical foundations. Technical Report MTR-2547, Vol. 1, MITRE Corp., Bedford, MA, 1973.

[4] Elisa Bertino, Piero Andrea Bonatti, and Elena Ferrari. Trbac: A temporal role-based access control model. *ACM Trans. Inf. Syst. Secur.*, 4(3):191–233, August 2001.

[5] J. Boender, G. Primiero, and Raimondi F. Minimizing transitive trust threats in software management systems. Technical report, Department of Computer Science, Middlesex University, 2014.

[6] Sudip Chakraborty and Indrajit Ray. Trustbac: Integrating trust relationships into the rbac model for access control in open systems. In *Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies*, SACMAT '06, pages 49–58, New York, NY, USA, 2006. ACM.

[7] S.J. Chandran and J. B. D. Joshi. Lot-rbac: A location and time-based rbac model. In Anne H. H. Ngu, Masaru Kitsuregawa, Erich J. Neuhold, Jen-Yao Chung, and Quan Z. Sheng, editors, *WISE*, volume 3806 of *Lecture Notes in Computer Science*, pages 361–375. Springer, 2005.

[8] Bruce Christianson. Living in an impossible world: Real-izing the consequences of intransitive trust. *Philosophy & Technology*, 26(4):411–429, 2013.

[9] Bruce Christianson and William S. Harbison. Why isn't trust transitive? In T. Mark A. Lomas, editor, *Security Protocols Workshop*, volume 1189 of *Lecture Notes in Computer Science*, pages 171–176. Springer, 1996.

[10] Stephen Clarke, Bruce Christianson, and Hannan Xiao. Trust*: Using local guarantees to extend the reach of trust. In Bruce Christianson, James A. Malcolm, Vashek Matyas, and Michael Roe, editors, *Security Protocols Workshop*, volume 7028 of *Lecture Notes in Computer Science*, pages 171–178. Springer, 2009.

[11] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli. Proposed nist standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4(3):224–274, August 2001.

[12] V. Genovese. *Modalities in Access Control: Logics, Proof-theory and Applications*. PhD thesis, University of Luxembourg and University of Torino, 2012.

[13] Audun Jøsang and Simon Pope. Semantic constraints for trust transitivity. In *Proceedings of the 2Nd Asia-Pacific Conference on Conceptual Modelling - Volume 43*, APCCM '05, pages 59–68, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc.

[14] Ueli M. Maurer and Pierre E. Schmid. A calculus for security bootstrapping in distributed systems. *Journal of Computer Security*, 4(1):55–80, 1996.

[15] Vladimir A. Oleshchuk. Trust-aware rbac. In Igor V. Kotenko and Victor A. Skormin, editors, *MMM-ACNS*, volume 7531 of *Lecture Notes in Computer Science*, pages 97–107. Springer, 2012.