Formal Definitions & Complexity Results for Trust Relations & Trust Domains

(Talk at Nanyang Technological University)

Simon Kramer
(j.w.w. Rajeev Goré, ANU & Eiji Okamoto, U Tsukuba)

University of Luxembourg
Institute of Mathematical Sciences, Chennai

April 1, 2011

Purpose of this talk

To present:

1. *formal definitions* for **trust relations** & **trust domains** that are *computational* & *declarative*
2. computational *complexity results* for deciding trust relationships & membership in trust domains
3. *compositionality* and *scalability results* for trust domains
4. instantiations of our trust concepts in 4 major *applications:*
   - 4.1 Trusted Third Parties (TTPs)
   - 4.2 the Web of Trust
   - 4.3 Public-Key Infrastructures (PKIs)
   - 4.4 Identity-Based Cryptography
5. computational means for *building trust*, and by that, building up trust relations & trust domains.

Functionality guarantee of dependable distributed systems

Dependable distributed systems guarantee their functionality:

	in spite of	**thanks to**
agents	*incorrect behaviour:* incorrect use of the technology due to unawareness and/or maliciousness	*correct behaviour:* correct use of the technology
technology	*incorrect design, natural forces*	*correct design*

Conditions on the functionality guarantee

The functionality guarantee is conditioned on:

1. **naming:** agent identity for the *information security* aspect [And08, Chapter 6] (anonymity, pseudonymity)
2. **number:** a minimal number of **correct** (or dually, a maximal number of faulty/corrupt) **agents** for the aspects of
   2.1 *fault tolerance* [Lyn96] (classical distributed computation)
   2.2 *corruption tolerance* [Yao82] (secure multiparty computation).

Notion of agent correctness

The notion of **agent correctness** (e.g., as induced by a *policy*) depends on the system itself.
For example, agent correctness can include:

1. algorithmic compliance
2. liveness (absence of crash)
3. fairness (scheduling of services)
4. **absence of cryptographic-key compromise**
5. etc.

Trust from knowledge of agent correctness

- Agent correctness captures the **predictability** of each correct partaking agent that guarantees the functionality of the system to all, correct or faulty, partaking agents.
- In sum, system functionality depends on agent correctness, and agents depend on each other via each other's correctness.
- Whence the **social need**, called **trust**, to *know whether or not someone is behaving correctly.*

  For example in: auctions, banking and other commerce, health care, social networking, voting, governmental administration, etc.

Aspects of trust

Trust has at least 3 aspects:

1. **trust relations:** An agent a may trust an agent b when a *believes or even knows that b behaves correctly.*
2. **trust domains:** A trust domain is *a community of mutually trusting agents with the common belief in or even knowledge of the trust relationships in the community.*

   Informally, a statement ϕ is **common belief or knowledge** in a community $\mathcal{C}$ when *all agents in $\mathcal{C}$ believe or know that ϕ is true (call this new statement ϕ'), all believe or know that ϕ' is true (call this new statement ϕ''), all believe or know that ϕ'' is true (call this new statement ϕ'''), etc. [HM90]*

Aspects of trust (continued)

3. **trust management** (cf. [RK05] for a recent survey):
   3.1 the *organisation of trust relations into trust domains*, i.e., the sociology.
   This requires the ability to decide whether or not a given:
   - relation is a trust relation
   - domain is a trust domain.

   If possible with the aid of a computer, cf. *Computer-Aided Decision Making (CADM)*; but then *we need formal definitions and decidability (and if possible, complexity) results!*
   3.2 the *coordination of trust-building actions*, i.e., the flow of trust (e.g., building reputation [sIB07]).

Goal

Our goal is five-fold, namely:

1. *formal definitions* for **trust relations** & **trust domains** that are *computational* & *declarative*
2. computational *complexity results* for deciding trust relationships & membership in trust domains
3. *compositionality* and *scalability results* for trust domains
4. instantiations of our trust concepts in 4 major *applications:*
   4.1 Trusted Third Parties (TTPs)
   4.2 the Web of Trust
   4.3 Public-Key Infrastructures (PKIs)
   4.4 Identity-Based Cryptography
5. computational means for *building trust*, and by that, building up trust relations & trust domains.

Methodology

Our methodology for meeting our goal is:

1. to develop our *formal definitions* for trust relations & trust domains
   - in a framework that is a semantically defined, standard **modal logic of belief & knowledge**
   - based on the **defining principle** of

   belief in or knowledge of agent correctness.
2. to derive our *complexity results* for deciding trust relationships & membership in trust domains **by reduction** to known results for the complexity of belief & knowledge.

Modal language

Let

- $\mathcal{A}$ designate an arbitrary finite set of *agent names* a, b, c, etc.
- $\mathcal{C} \subseteq \mathcal{A}$ denote (finite and not necessarily disjoint) communities of agents (referred to by their name)
- $\mathcal{P} := \{ \text{correct}(a) \mid a \in \mathcal{A} \}$ designate our (finite) set of *atomic propositions* P for referring to **agent correctness**
- $\mathcal{L} \ni \phi ::= P \mid \neg\phi \mid \phi \wedge \phi \mid \mathsf{CB}_{\mathcal{C}}(\phi) \mid \mathsf{CK}_{\mathcal{C}}(\phi)$ designate our **modal language** of formulae ϕ, with $\mathsf{CB}_{\mathcal{C}}(\phi)$ for "it is *common belief* in the community $\mathcal{C}$ that ϕ", and $\mathsf{CK}_{\mathcal{C}}(\phi)$ for "it is *common knowledge* in the community $\mathcal{C}$ that ϕ".

Modal model

Let

- $\mathcal{S}$ designate a set (the *state space*) of *system states* s
- $\mathrm{D}_a \subseteq \mathcal{S} \times \mathcal{S}$ designate *serial*, *transitive*, and *Euclidean* relations of *doxastic accessibility*
- $\mathrm{E}_a \subseteq \mathcal{S} \times \mathcal{S}$ designate *equivalence* relations of *epistemic accessibility* (e.g., state *indistinguishability*)
- $\mathfrak{S} := (\mathcal{S}, \{\mathrm{D}_a\}_{a\in\mathcal{A}}, \{\mathrm{E}_a\}_{a\in\mathcal{A}})$ designate the (modal) *frame* of our framework such that $\mathrm{D}_a \subseteq \mathrm{E}_a$ for any $a \in \mathcal{A}$
- $\mathcal{V} : \mathcal{P} \to 2^{\mathcal{S}}$ designate a *valuation function*
- $(\mathfrak{S}, \mathcal{V})$ designate the (modal) **model.**

Satisfaction relation

Let

- $\mathrm{D}^{+}_{\mathcal{C}}$ designate the *transitive* closure of $\bigcup_{a\in\mathcal{C}} \mathrm{D}_a$
- $\mathrm{E}^{*}_{\mathcal{C}}$ designate the *reflexive transitive* closure of $\bigcup_{a\in\mathcal{C}} \mathrm{E}_a$.

Then the **satisfaction relation** $\models$ of our framework is:

$$\begin{array}{rcl}
(\mathfrak{S}, \mathcal{V}), s \models P & :\text{iff} & s \in \mathcal{V}(P) \\
(\mathfrak{S}, \mathcal{V}), s \models \neg\phi & :\text{iff} & \text{not } (\mathfrak{S}, \mathcal{V}), s \models \phi \\
(\mathfrak{S}, \mathcal{V}), s \models \phi \wedge \phi' & :\text{iff} & (\mathfrak{S}, \mathcal{V}), s \models \phi \text{ and } (\mathfrak{S}, \mathcal{V}), s \models \phi' \\
(\mathfrak{S}, \mathcal{V}), s \models \mathsf{CB}_{\mathcal{C}}(\phi) & :\text{iff} & \text{for all } s' \in \mathcal{S}, \text{ if } s\ \mathrm{D}^{+}_{\mathcal{C}}\ s' \text{ then } (\mathfrak{S}, \mathcal{V}), s' \models \phi \\
(\mathfrak{S}, \mathcal{V}), s \models \mathsf{CK}_{\mathcal{C}}(\phi) & :\text{iff} & \text{for all } s' \in \mathcal{S}, \text{ if } s\ \mathrm{E}^{*}_{\mathcal{C}}\ s' \text{ then } (\mathfrak{S}, \mathcal{V}), s' \models \phi.
\end{array}$$

Weak & strong trust relations & domains

Let $\phi \vee \phi' := \neg(\neg\phi \wedge \neg\phi')$, $\top := \mathsf{correct}(a) \vee \neg\mathsf{correct}(a)$, $\bot := \neg\top$, $\phi \to \phi' := \neg\phi \vee \phi'$, $\phi \leftrightarrow \phi' := (\phi \to \phi') \wedge (\phi' \to \phi)$, $\mathsf{B}_a(\phi) := \mathsf{CB}_{\{a\}}(\phi)$ (for "a believes that ϕ"), and $\mathsf{K}_a(\phi) := \mathsf{CK}_{\{a\}}(\phi)$ (for "a knows that ϕ").

Then we can define **(dis)trust relations** & **(dis)trust domains:**

a wTrusts b	$:=$	$\mathsf{B}_a(\mathsf{correct}(b))$	a weakly trusts b
$\mathsf{wTD}(\mathcal{C})$	$:=$	$\mathsf{CB}_{\mathcal{C}}(\bigwedge_{a,b\in\mathcal{C}} a \text{ wTrusts } b)$	$\mathcal{C}$ is a weak TD
a sTrusts b	$:=$	$\mathsf{K}_a(\mathsf{correct}(b))$	a strongly trusts b
$\mathsf{sTD}(\mathcal{C})$	$:=$	$\mathsf{CK}_{\mathcal{C}}(\bigwedge_{a,b\in\mathcal{C}} a \text{ sTrusts } b)$	$\mathcal{C}$ is a strong TD.

Truth & Validity

- The formula ϕ is *true* (or *satisfied*) in the model $(\mathfrak{S}, \mathcal{V})$ at the state $s \in \mathcal{S}$:iff $(\mathfrak{S}, \mathcal{V}), s \models \phi$.
- The formula ϕ is *satisfiable* in the model $(\mathfrak{S}, \mathcal{V})$:iff there is $s \in \mathcal{S}$ such that $(\mathfrak{S}, \mathcal{V}), s \models \phi$.
- The formula ϕ is *globally true* (or *globally satisfied*) in the model $(\mathfrak{S}, \mathcal{V})$, written $(\mathfrak{S}, \mathcal{V}) \models \phi$, :iff for all $s \in \mathcal{S}$, $(\mathfrak{S}, \mathcal{V}), s \models \phi$.
- The formula ϕ is *satisfiable* :iff there is a model $(\mathfrak{S}, \mathcal{V})$ and a state $s \in \mathcal{S}$ such that $(\mathfrak{S}, \mathcal{V}), s \models \phi$.
- The formula ϕ is *valid*, written $\models \phi$, :iff for all models $(\mathfrak{S}, \mathcal{V})$, $(\mathfrak{S}, \mathcal{V}) \models \phi$ (cf. [BvB07]).

Properties of common belief

1. $\models \mathsf{CB}_{\mathcal{C}}(\phi \to \phi') \to (\mathsf{CB}_{\mathcal{C}}(\phi) \to \mathsf{CB}_{\mathcal{C}}(\phi'))$ (Kripke's law)
2. $\models \mathsf{CB}_{\{a\}}(\phi) \to \neg\mathsf{CB}_{\{a\}}(\neg\phi)$ (consistency of beliefs)
3. $\models \mathsf{CB}_{\mathcal{C}}(\phi) \to \mathsf{CB}_{\mathcal{C}}(\mathsf{CB}_{\mathcal{C}}(\phi))$ (positive introspection)
4. $\models \neg\mathsf{CB}_{\mathcal{C}}(\phi) \to \mathsf{CB}_{\mathcal{C}}(\neg\mathsf{CB}_{\mathcal{C}}(\phi))$ (negative introspection)
5. if $\models \phi$ then $\models \mathsf{CB}_{\mathcal{C}}(\phi)$ (necessitation)
6. $\models \mathsf{CB}_{\mathcal{C}}(\phi) \to \mathsf{EB}_{\mathcal{C}}(\phi)$
7. $\models \mathsf{CB}_{\mathcal{C}}(\phi) \to \mathsf{EB}_{\mathcal{C}}(\mathsf{CB}_{\mathcal{C}}(\phi))$
8. $\models \mathsf{CB}_{\mathcal{C}}(\phi \to \mathsf{EB}_{\mathcal{C}}(\phi)) \to (\mathsf{EB}_{\mathcal{C}}(\phi) \to \mathsf{CB}_{\mathcal{C}}(\phi))$,

where $\mathsf{EB}_{\mathcal{C}}(\phi) := \bigwedge_{a\in\mathcal{C}} \mathsf{B}_a(\phi)$ (cf. [MV07, Section 7.1]).

Outline
Introduction
Formal definitions
Complexity results
Application to cryptographic-key management
Conclusion
Bibliography
Framework
Concepts & properties

Properties of common knowledge

1. $\models \mathsf{CK}_{\mathcal{C}}(\phi \to \phi') \to (\mathsf{CK}_{\mathcal{C}}(\phi) \to \mathsf{CK}_{\mathcal{C}}(\phi'))$ (Kripke's law)
2. $\models \mathsf{CK}_{\mathcal{C}}(\phi) \to \phi$ (truth law)
3. $\models \mathsf{CK}_{\mathcal{C}}(\phi) \to \mathsf{CK}_{\mathcal{C}}(\mathsf{CK}_{\mathcal{C}}(\phi))$ (positive introspection)
4. $\models \neg\mathsf{CK}_{\mathcal{C}}(\phi) \to \mathsf{CK}_{\mathcal{C}}(\neg\mathsf{CK}_{\mathcal{C}}(\phi))$ (negative introspection)
5. if $\models \phi$ then $\models \mathsf{CK}_{\mathcal{C}}(\phi)$ (necessitation)
6. $\models \mathsf{CK}_{\mathcal{C}}(\phi) \to \mathsf{EK}_{\mathcal{C}}(\mathsf{CK}_{\mathcal{C}}(\phi))$
7. $\models \mathsf{CK}_{\mathcal{C}}(\phi \to \mathsf{EK}_{\mathcal{C}}(\phi)) \to (\phi \to \mathsf{CK}_{\mathcal{C}}(\phi))$,

where $\mathsf{EK}_{\mathcal{C}}(\phi) := \bigwedge_{a\in\mathcal{C}} \mathsf{K}_a(\phi)$ (cf. [MV07, Section 7.1]).

Simon Kramer (j.w.w. Rajeev Goré, ANU & Eiji Okamoto, U T Trust Relations & Domains: Definitions & Complexities

Outline
Introduction
Formal definitions
Complexity results
Application to cryptographic-key management
Conclusion
Bibliography
Framework
Concepts & properties

Properties of trust

1. **Belief versus knowledge:** For all $\mathcal{C} \subseteq \mathcal{A}$,
   $\models \mathsf{CK}_{\mathcal{C}}(\phi) \to \mathsf{CB}_{\mathcal{C}}(\phi)$.
   In particular when $\mathcal{C} = \{a\}$, $\models \mathsf{K}_a(\phi) \to \mathsf{B}_a(\phi)$.
2. **Strong versus weak trust:**
   2.1 For all $a, b \in \mathcal{A}$, $\models a \,\mathsf{sTrusts}\, b \to a \,\mathsf{wTrusts}\, b$.
   2.2 For all $\mathcal{C} \subseteq \mathcal{A}$, $\models \mathsf{sTD}(\mathcal{C}) \to \mathsf{wTD}(\mathcal{C})$.
3. In trust domains, trust relations are *universal* (i.e., correspond to the Cartesian product on those domains). That is, for all $\mathcal{C} \subseteq \mathcal{A}$ and $a, b \in \mathcal{C}$:

$$\models \mathsf{wTD}(\mathcal{C}) \to a \,\mathsf{wTrusts}\, b \qquad \models \mathsf{sTD}(\mathcal{C}) \to a \,\mathsf{sTrusts}\, b.$$

Simon Kramer (j.w.w. Rajeev Goré, ANU & Eiji Okamoto, U T Trust Relations & Domains: Definitions & Complexities

Outline
Introduction
Formal definitions
Complexity results
Application to cryptographic-key management
Conclusion
Bibliography
Framework
Concepts & properties

Properties of trust relations

Simple corollary: in trust domains, trust relations are:

1. *reflexive*. That is, for all $\mathcal{C} \subseteq \mathcal{A}$ and $a \in \mathcal{C}$:

$$\models \mathsf{wTD}(\mathcal{C}) \to a \,\mathsf{wTrusts}\, a \qquad \models \mathsf{sTD}(\mathcal{C}) \to a \,\mathsf{sTrusts}\, a.$$

2. *symmetric*. That is, for all $\mathcal{C} \subseteq \mathcal{A}$ and $a, b \in \mathcal{C}$:

$$\models \mathsf{wTD}(\mathcal{C}) \to (a \,\mathsf{wTrusts}\, b \to b \,\mathsf{wTrusts}\, a)$$
$$\models \mathsf{sTD}(\mathcal{C}) \to (a \,\mathsf{sTrusts}\, b \to b \,\mathsf{sTrusts}\, a).$$

3. *transitive*. That is, for all $\mathcal{C} \subseteq \mathcal{A}$ and $a, b, c \in \mathcal{C}$:

$$\models \mathsf{wTD}(\mathcal{C}) \to ((a \,\mathsf{wTrusts}\, b \,\wedge\, b \,\mathsf{wTrusts}\, c) \to a \,\mathsf{wTrusts}\, c)$$
$$\models \mathsf{sTD}(\mathcal{C}) \to ((a \,\mathsf{sTrusts}\, b \,\wedge\, b \,\mathsf{sTrusts}\, c) \to a \,\mathsf{sTrusts}\, c).$$

Simon Kramer (j.w.w. Rajeev Goré, ANU & Eiji Okamoto, U T Trust Relations & Domains: Definitions & Complexities

Properties of trust relations (continued)

Lemma (Transitivity Lemma)

Let $a, b \in \mathcal{A}$. Then for all $c \in \mathcal{A}$:

1. $\models \mathsf{B}_a(c \text{ sTrusts } b) \rightarrow a \text{ wTrusts } b$
2. $\models \mathsf{K}_a(c \text{ sTrusts } b) \rightarrow a \text{ sTrusts } b$.

c acts as a *reference* of b's trustworthiness to a — important for applications, e.g., for the construction of (transitive) **trust paths.**

Properties of trust domains

0. $\models \mathsf{wTD}(\emptyset)$ and $\models \mathsf{sTD}(\emptyset)$
1. Separating a trust domain:
   $\models \mathsf{wTD}(\mathcal{C} \cup \mathcal{C}') \rightarrow (\mathsf{wTD}(\mathcal{C}) \wedge \mathsf{wTD}(\mathcal{C}'))$
   $\models \mathsf{sTD}(\mathcal{C} \cup \mathcal{C}') \rightarrow (\mathsf{sTD}(\mathcal{C}) \wedge \mathsf{sTD}(\mathcal{C}'))$
2. $\models (\mathsf{wTD}(\mathcal{C}) \wedge \mathsf{wTD}(\mathcal{C}')) \rightarrow \mathsf{wTD}(\mathcal{C} \cap \mathcal{C}')$
   $\models (\mathsf{sTD}(\mathcal{C}) \wedge \mathsf{sTD}(\mathcal{C}')) \rightarrow \mathsf{sTD}(\mathcal{C} \cap \mathcal{C}')$
3. if $\mathcal{C} \subseteq \mathcal{C}'$ then $\models \mathsf{wTD}(\mathcal{C}') \rightarrow \mathsf{wTD}(\mathcal{C})$ and $\models \mathsf{sTD}(\mathcal{C}') \rightarrow \mathsf{sTD}(\mathcal{C})$.

Properties of trust domains (continued)

Theorem (Merging strong trust domains)

Merging two strong trust domains is compositional in the sense that a necessary and sufficient condition for merging two strong trust domains is that it be common knowledge in the union of both domains that each domain is a strong trust domain. Formally, for all $\mathcal{C}, \mathcal{C}' \subseteq \mathcal{A}$,

$$\models \mathsf{CK}_{\mathcal{C} \cup \mathcal{C}'}(\mathsf{sTD}(\mathcal{C}) \wedge \mathsf{sTD}(\mathcal{C}')) \leftrightarrow \mathsf{sTD}(\mathcal{C} \cup \mathcal{C}').$$

Building up trust via a recursive descent RD

1. **Input:** a model $(\mathfrak{S}, \mathcal{V})$, a state s, and $\mathcal{C}_1, \mathcal{C}_2 \in \mathcal{A}$;
2. **Divide:** for $i \in \{1, 2\}$ do {
   when $(\mathfrak{S}, \mathcal{V}), s \models \neg\mathsf{sTD}(\mathcal{C}_i)$:
   2.1 divide $\mathcal{C}_i$ freely into $\mathcal{C}_{i.1}$ and $\mathcal{C}_{i.2}$;
   2.2 $s := \mathrm{RD}((\mathfrak{S}, \mathcal{V}), s, \mathcal{C}_{i.1}, \mathcal{C}_{i.2})$; };
3. **Conquer:** announce to the community $\mathcal{C}_1 \cup \mathcal{C}_2$ that $\mathsf{sTD}(\mathcal{C}_1) \wedge \mathsf{sTD}(\mathcal{C}_2)$ is true (choose appropriate communication channels and an appropriate protocol), which takes the system from s to some $s' \in \mathcal{S}$ such that s' is reachable from s and
   $$(\mathfrak{S}, \mathcal{V}), s' \models \mathsf{CK}_{\mathcal{C} \cup \mathcal{C}'}(\mathsf{sTD}(\mathcal{C}) \wedge \mathsf{sTD}(\mathcal{C}'));$$
4. **Output:** $s' \in \mathcal{S}$;
5. **Effect:** $(\mathfrak{S}, \mathcal{V}), s' \models \mathsf{sTD}(\mathcal{C}_1 \cup \mathcal{C}_2)$.

Potential trust

The idea is to define **potentiality** as satis**fiability**.

- There is a potential weak (strong) trust relationship between a and b in the model $(\mathfrak{S}, \mathcal{V})$:iff a wTrusts b (a sTrusts b) is satisfiable in $(\mathfrak{S}, \mathcal{V})$.
- The community $\mathcal{C}$ is a potential weak (strong) trust domain in the model $(\mathfrak{S}, \mathcal{V})$:iff wTD($\mathcal{C}$) (sTD($\mathcal{C}$)) is satisfiable in $(\mathfrak{S}, \mathcal{V})$.

Similarly, we define *actual* trust between two agents.
The idea is to define (two degrees of) **actuality** as (two degrees of) satis**faction.**

Actual trust

- There is a weak (strong) trust relationship between a and b in the model $(\mathfrak{S}, \mathcal{V})$ at the state $s \in \mathcal{S}$:iff a wTrusts b (a sTrusts b) is satisfied in $(\mathfrak{S}, \mathcal{V})$ at s.
- There is a weak (strong) trust relationship between a and b in the model $(\mathfrak{S}, \mathcal{V})$:iff a wTrusts b (a sTrusts b) is globally satisfied in $(\mathfrak{S}, \mathcal{V})$.
- The community $\mathcal{C}$ is a weak (strong) trust domain in the model $(\mathfrak{S}, \mathcal{V})$ at the state $s \in \mathcal{S}$:iff wTD($\mathcal{C}$) (sTD($\mathcal{C}$)) is satisfied in $(\mathfrak{S}, \mathcal{V})$ at s.
- The community $\mathcal{C}$ is a weak (strong) trust domain in the model $(\mathfrak{S}, \mathcal{V})$:iff wTD($\mathcal{C}$) (sTD($\mathcal{C}$)) is globally satisfied in $(\mathfrak{S}, \mathcal{V})$.

Potential versus actual trust

- Since satisfaction implies satisfiability, but not vice versa, actual trust implies potential trust, but not vice versa.
- For example, if two agents do not even know each other then they can not be in an actual trust relationship.
- However, they may be in a potential trust relationship: maybe in another system state, their trust potential can become actualised.
- On the other hand, in a given system two agents may well know each other but not be in a potential trust relationship: **the system may be designed so that trust between them is impossible—in any system state.**

Computational time complexities

Assumption. The truth of each atomic proposition can be deterministically decided in a polynomial number $f(|s|)$ of steps in the size $|s|$ of the state s of the model $(\mathfrak{S}, \mathcal{V})$.

Then, we have the following results:

		Trust relations		Trust domains	
	degree	*weak* a wTrusts b	*strong* a sTrusts b	*weak* wTD($\mathcal{C}$)	*strong* sTD($\mathcal{C}$)
actual	local satisfaction	$\mathcal{O}(f(\lvert s\rvert))$		$\mathcal{O}(f(\lvert s\rvert) \cdot 2^{\lvert\mathcal{C}\rvert})$	
	global satisfaction				
potential	satisfiability				

Computationally speaking, **collective trust does not scale.**
Trust domains should be family-sized, so to speak.

Application of our framework

- **What it means:** to define the valuation function $\mathcal{V}$ on the atomic propositions correct(a) about agent correctness.
- Each notion of agent correctness is specific to each system rather than general to all systems.
- **Trust is system-specific to some extent.**
- **However:** we can define agent correctness *generically* for the Web of Trust & PKIs, by means of a common auxiliary logic, called AuxLog.

Trustworthy Trusted Third Parties

TTPs that may or even must *deserve* the trust of their trusters—and vice versa:

- c is a weakly trustworthy TTP of a and b:

  $$\mathrm{wtTTP}(c,a,b) := \mathrm{CB}_{\{a,b,c\}}(\mathrm{wTD}(\{c,a\}) \land \mathrm{wTD}(\{c,b\}))$$

- c is a strongly trustworthy TTP of a and b:

  $$\mathrm{stTTP}(c,a,b) := \mathrm{CK}_{\{a,b,c\}}(\mathrm{sTD}(\{c,a\}) \land \mathrm{sTD}(\{c,b\}))$$

The two sides (e.g., $\{c,a\}$ and $\{c,b\}$) in a strongly (but not in a weakly) trustworthy TTP constitute a (strong) trust domain as a whole (i.e., as $\{c,a\} \cup \{c,b\}$)!

Auxiliary logic

AuxLog is *parametric* in a binary (constant-time decidable) relation $R \subseteq \mathcal{A} \times \mathcal{A}$ to be fixed separately for the Web of Trust & PKIs.

Let

- $\mathcal{X}$ designate a countable set of propositional variables C
- $\mathcal{L}' \ni \alpha ::= \mathrm{OK} \mid C \mid \neg\alpha \mid \alpha \land \alpha \mid \overline{\Box}\alpha \mid \boldsymbol{\nu} C(\alpha)$ designate the language $\mathcal{L}'$ of AuxLog where all free occurrences of C in α of $\boldsymbol{\nu} C(\alpha)$ are assumed to occur within an even number of occurrences of $\neg$ to guarantee the existence of (greatest) fixpoints (expressed by $\boldsymbol{\nu} C(\alpha)$) [BS07].

Auxiliary logic (continued)

Then, given an auxiliary interpretation $[\![\cdot]\!] : \mathcal{X} \cup \{\mathrm{OK}\} \to 2^{\mathcal{A}}$ s.t.

$$[\![\mathrm{OK}]\!] := \{\ a \in \mathcal{A} \mid \text{at most } a \text{ can access } a\text{'s private key}\ \},$$

the interpretation $\|\cdot\|_{[\![\cdot]\!]} : \mathcal{L}' \to 2^{\mathcal{A}}$ of AuxLog-propositions is:

$$\begin{aligned}
\|X\|_{[\![\cdot]\!]} &:= [\![X]\!], \text{ where } X \in \mathcal{X} \cup \{\mathrm{OK}\} \\
\|\neg\alpha\|_{[\![\cdot]\!]} &:= \mathcal{A} \setminus \|\alpha\|_{[\![\cdot]\!]} \\
\|\alpha \land \alpha'\|_{[\![\cdot]\!]} &:= \|\alpha\|_{[\![\cdot]\!]} \cap \|\alpha'\|_{[\![\cdot]\!]} \\
\|\overline{\Box}\alpha\|_{[\![\cdot]\!]} &:= \{\ a \in \mathcal{A} \mid \text{for all } b \in \mathcal{A}, \text{ if } b\, R\, a \text{ then } b \in \|\alpha\|_{[\![\cdot]\!]}\ \} \\
\|\boldsymbol{\nu} C(\alpha)\|_{[\![\cdot]\!]} &:= \bigcup\{\ A \subseteq \mathcal{A} \mid A \subseteq \|\alpha\|_{[\![\cdot]\!]_{[C\mapsto A]}}\ \},
\end{aligned}$$

where $[\![\cdot]\!]_{[C\mapsto A]}$ maps C to A and otherwise agrees with $[\![\cdot]\!]$.

Auxiliary logic (end)

Further, $\alpha \vee \alpha' := \neg(\neg\alpha \wedge \neg\alpha')$, $\top := \alpha \vee \neg\alpha$, $\bot := \neg\top$, $\alpha \to \alpha' := \neg\alpha \vee \alpha'$, $\alpha \leftrightarrow \alpha' := (\alpha \to \alpha') \wedge (\alpha' \to \alpha)$, $\overline{\Diamond}\alpha := \neg\overline{\Box}(\neg\alpha)$, and, notably, $\boldsymbol{\mu} C(\alpha(C)) := \neg\boldsymbol{\nu} C(\neg\alpha(\neg C))$.

Finally, for all $a \in \mathcal{A}$ and $\alpha \in \mathcal{L}'$,

$$\langle(\mathcal{A}, R), [\![\cdot]\!]\rangle, a \vDash \alpha \quad \text{:iff} \quad a \in \|\alpha\|_{[\![\cdot]\!]},$$

Outline
Introduction
Formal definitions
Complexity results
Application to cryptographic-key management
Conclusion
Bibliography
Trusted Third Parties
The Web of Trust
Public-Key Infrastructures
Identity-Based Cryptography

The Web of Trust(ed introducers)

- The role of an agent a's **trusted introducer** b is to act as a *guarantor* for the trustworthiness of a, and by that, to catalyse the building up of trust relationships between a and those agents c who are only potential (not yet actual) trustees of a but who are (already) actual trustees of b.
- Agents are (socially speaking) trustworthy, or (technically speaking) **correct if and only if all their designated trusted introducers are, and at most they (the correct agents) can access their (own) private key.**
  (Agents with untrustworthy introducers or a corrupt private key are untrustworthy.)

Simon Kramer (j.w.w. Rajeev Goré, ANU & Eiji Okamoto, U T Trust Relations & Domains: Definitions & Complexities

Outline
Introduction
Formal definitions
Complexity results
Application to cryptographic-key management
Conclusion
Bibliography
Trusted Third Parties
The Web of Trust
Public-Key Infrastructures
Identity-Based Cryptography

The Web of Trust (continued)

- We model the designated-trusted-introducer relationships between agents in system states $s \in \mathcal{S}$ with a family of relations $\mathrm{DTI}_s \subseteq \mathcal{A} \times \mathcal{A}$ such that

  $b \; \mathrm{DTI}_s \; a$:iff b is a designated trusted introducer of a in s.

- The valuation function $\mathcal{V}$ on the propositions correct(a) can then be formally defined with the aid of AuxLog as:

  $$\boxed{\mathcal{V}(\mathsf{correct}(a)) := \{\; s \mid \langle(\mathcal{A}, \mathrm{DTI}_s), \emptyset\rangle, a \vDash \boldsymbol{\nu} C(\mathsf{OK} \wedge \overline{\Box} C) \;\},}$$

  where $\emptyset$ designates the empty auxiliary interpretation (C is bound!).

Simon Kramer (j.w.w. Rajeev Goré, ANU & Eiji Okamoto, U T Trust Relations & Domains: Definitions & Complexities

Outline
Introduction
Formal definitions
Complexity results
Application to cryptographic-key management
Conclusion
Bibliography
Trusted Third Parties
The Web of Trust
Public-Key Infrastructures
Identity-Based Cryptography

The Web of Trust (end)

This *co*-inductive definition has the following iterative paraphrase from *above* (iterated *de*construction).

Everybody is correct (the Web of Trust is born in the plenum, so to say); except for the following agents (exclude those which are clearly not OK):

0. *agents with a corrupt private key (Type 0 agents)*
1. *agents with a designated trusted introducer of Type 0 (Type 1 agents)*
2. *agents with a designated trusted introducer of Type 1 (Type 2 agents)*
3. *etc.*

Simon Kramer (j.w.w. Rajeev Goré, ANU & Eiji Okamoto, U T Trust Relations & Domains: Definitions & Complexities

Public-Key Infrastructures & Certificate Authorities

- Centralised **certificate authorities** (CAs) act as guarantors for the trustworthiness of the public key of their clients by issuing certificates that bind the public-key of each client (the key owner) to the client's (unique) name.
- Agents are (socially speaking) trustworthy, or (technically speaking) **correct if and only if all their certified agents are, and at most they (the correct agents) can access their (own) private key.**
  (Agents who certify incorrect agents or with corrupt private keys are incorrect.)

Public-Key Infrastructures & CAs (continued)

- We model the relationships from certifying agents to certified agents in system states $s \in \mathcal{S}$ with a family of relations $\mathrm{CRT}_s \subseteq \mathcal{A} \times \mathcal{A}$ such that

  $$b \; \mathrm{CRT}_s \; a \text{ :iff } b \text{ is certified by } a \text{ in } s,$$

  where "b is certified by a in s" means "a has issued a valid certificate for b in s", i.e., a certificate that is non-revoked in s and signed by a with the private key of a.
- The valuation function $\mathcal{V}$ on the propositions correct(a) can then be formally defined with the aid of AuxLog as

$$\mathcal{V}(\mathrm{correct}(a)) := \{ \, s \mid \langle(\mathcal{A}, \mathrm{CRT}_s), \emptyset\rangle, a \vDash \mu C(\mathsf{OK} \land \overline{\Box} C) \, \}.$$

Public-Key Infrastructures & CAs (continued)

This *in*ductive definition has the following iterative paraphrase from *below* (iterated *con*struction).

> *Nobody is correct (PKIs are born ex nihilo, so to say); except for the following agents (include those which are clearly OK): agents without a corrupt private key (Type 0 agents), whose certified agents are also of Type 0 (Type 1 agents), whose certified agents are again also of Type 0 (Type 2 agents), etc.*
> *(In other words, being of Type 0 is an invariant in the transitive closure of certification relationships.)*

Public-Key Infrastructures & CAs (end)

- CAs are commonly organised in a hierarchy, which induces **structured trust domains** in the form of finite trees $\leq$.
- Trust relations are *symmetric* (up- and downwards the tree branches) and *transitive* (along the tree branches).
- Hence we can fit our weak and strong trust domains to PKI trust domains with a simple constraint:

$$\mathsf{wTD}_{\mathsf{PKI}}(\mathcal{C}) := \mathsf{CB}_{\mathcal{C}}\Big(\bigwedge\nolimits_{a,\, b \,\in\, \mathcal{C} \text{ and } (a \leq b \text{ or } b \leq a)} a \; \mathsf{wTrusts} \; b\Big)$$
$$\mathsf{sTD}_{\mathsf{PKI}}(\mathcal{C}) := \mathsf{CK}_{\mathcal{C}}\Big(\bigwedge\nolimits_{a,\, b \,\in\, \mathcal{C} \text{ and } (a \leq b \text{ or } b \leq a)} a \; \mathsf{sTrusts} \; b\Big)$$

The case of Identity-Based Cryptography

- The intending sender of a message derives the (public) encryption key from the public identity (e.g., phone number) of the intended recipient [JN09].
- The private key must not be derivable from its corresponding public counterpart without an additional trap-door information owned by a central CA (cCA $\in \mathcal{A}$). (ID-based domains have a star structure.)
- Hence, the definition of an agent being OK becomes

$$[\![\mathrm{OK}]\!] := \{\, a \in \mathcal{A} \mid \text{at most } a \text{ and cCA can access } a\text{'s private key} \,\}.$$

- **The notion of trust is weakened!**

Related work

There is a huge literature on notions of trust that are not formal in the sense of formal languages and semantics, and also on trust management, which however is not the subject matter of this work.

Formal works (loosely) related to ours are discussed in [KGO10].

To our knowledge, complexity results for deciding trust relations and trust domains, building up trust domains, and (non-)compositionality and non-scalability results for (weak) strong trust domains are novel.

Assessment

- We have provided simple, smooth definitions and complexity results for multi-agent trust in a single, standard framework:
  1. *weak* & *strong* trust relations & trust domains
  2. *potential* & *actual* trust relationships & membership in trust domains
- All our definitions are:
  1. *declarative* & *computational*
  2. *parametric* in an application-specific notion of agent correctness.
- We have validated our approach in 4 major applications of cryptographic-key management [KGO10].

Future work

- **Graded trust** relations and domains (knowledge is belief with 100% certitude)
- **Time:** study the evolution of the quality and quantity of trust in a given distributed system
- **Trust management:** build actual systems that
  1. build trust *from absence of trust*
  2. *re*build trust from *dis*trust.

R. Anderson.
Security Engineering: A Guide to Building Dependable Distributed Systems.
Wiley, second edition, 2008.

J. Bradfield and C. Stirling.
Handbook of Modal Logic, chapter Modal Mu-Calculi.
Volume 3 of Blackburn et al. [BvBW07], 2007.

P. Blackburn and J. van Benthem.
Handbook of Modal Logic, chapter Modal Logic: A Semantic Perspective.
Volume 3 of Blackburn et al. [BvBW07], 2007.

P. Blackburn, J. van Benthem, and F. Wolter, editors.

Handbook of Modal Logic, volume 3 of *Studies in Logic and Practical Reasoning*.
Elsevier, 2007.

J.Y. Halpern and Y. Moses.
Knowledge and common knowledge in a distributed environment.
Journal of the ACM, 37(3), 1990.

M. Joye and G. Neven.
Identity-Based Cryptography.
IOS Press, 2009.

S. Kramer, R. Goré, and E. Okamoto.
Formal definitions and complexity results for trust relations and trust domains.

In *Proceedings of the ESSLLI-affiliated Workshop on Logics in Security*, 2010.

N.A. Lynch.
Distributed Algorithms.
Morgan Kaufmann Publishers, 1996.

J.-J. Meyer and F. Veltnam.
Handbook of Modal Logic, chapter Intelligent Agents and Common Sense Reasoning.
Volume 3 of Blackburn et al. [BvBW07], 2007.

S. Ruohomaa and L. Kutvonen.
Trust management survey.
In *Proceedings of the Conference on Trust Management*, volume 3477 of *LNCS*. Springer, 2005.

A. Jøsang, R. Ismail, and C. Boyd.
A survey of trust and reputation systems for online service provision.
Decision Support Systems, 43(2), 2007.

A. Yao.
Protocols for secure computations.
In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, 1982.

Contact

Email:
simon.kramer@a3.epfl.ch

Homepage:
http://www.cipher.risk.tsukuba.ac.jp/~kramer/