TRUTHFUL MECHANISMS FOR BUILDING TRUST IN E-COMMERCE

Giovanna Melideo[1] and Guido Proietti[1,2]

[1] *Dipartimento di Informatica, Università di L'Aquila, Via Vetoio, 67010 L'Aquila, Italy*

[2] *Istituto di Analisi dei Sistemi ed Informatica, CNR, Viale Manzoni 30, 00185 Roma, Italy*

{melideo,proietti}@di.univaq.it

Abstract

A fundamental issue for a real uptake of commercial transactions over the web regards *trust* among the transacting entities, frequently unknown to each other. One solution to increase confidence in transactions is to use a network of TSPs (Trust Service Providers), called a *trust web*, which are third parties known and trusted by both entities, and an algorithm that establishes a trust path before carrying-out any e-commerce transaction.

In this paper we study the problem of building trust paths linking an entity initiating a transaction to a set of final merchants in a trust web from a "mechanism design" point of view. Namely, we consider TSPs as *strategic agents* which respond to incentives and may deviate from the protocol for a tangible gain. A *truthful* mechanism should define both the protocol and a suitable payment rule such that each agent maximizes her own utility when not deviating from the protocol, regardless of what other agents do.

We first address the problem from a "protocol design" perspective and, assuming that TSPs are *honest/obedient*, we propose a distributed search algorithm based on a *probabilistic* trust degree model [Mau96, DIM02] which generalizes that based only on boolean trust relationships proposed in [Ati02], and reduces the search space complexity by pruning the alternatives that do not satisfy (besides cost constraints) trust degree constraints (e.g., the "transitive" degree of trust accumulated along the path has to be greater than a given threshold value).

Then, when considering TSPs as *strategic agents*, we use this algorithm as a substrate to define truthful mechanisms for building suitable trust paths. Indeed, the main scope of this paper is to provide an answer to the following fundamental problem: does a payment function exist for the described problem such that the the resulting mechanism is truthful? By applying recent results appeared in [MPPW+04] we provide both positive and negative answers, depending on which constraint we add/drop and on which parameters are considered as a private information of agents.

E. Nardelli et al. (eds.), *Certification and Security in Inter-Organizational E-Service*

Keywords: Algorithmic Mechanism Design, Algorithms for the Internet, Game Theory, Trust, e-Commerce, Security Services, Public Key Certification

1. Introduction

Internet services are increasingly being used in daily life for e-commerce, web-access to information and inter-personal interactions via e-mail, but there is still major concern about the trustworthiness of these e-services.

Trust is a fundamental component in every business transaction. Customers must trust that merchants will provide the services that they advertise, will not disclose nor misuse private customer information such as its credit card number. Trust in the supplier's competence and honesty will influence the customer's decision as to which supplier to use. Merchants must trust that the buyer is able to pay for goods or services. Hence, for e-commerce to achieve the same levels of acceptance as traditional commerce, trust management has to be an intrinsic part of e-commerce.

More and more often, to increase confidence in commercial transactions over the Web, where the transacting parts are frequently unknown to each other, the design of new protocol is based on the enlistment of a third party, referred to as a *trust service provider* (TSP), acting as a trusted intermediary which assumes responsibility for a smooth transaction. TSPs are known and trusted by both customer and merchant.

Following [Ati02, CAR00, HFH99, DIM02], from a graph-theoretical point of view such a network of trusted intermediaries (referred to as *trust web*) can be modelled as a *trust graph*, where vertices denote the TSPs and edges reflet trust relationships between TSPs. Each TSP only knows its immediate trusted neighbors rather than each TSP in the global trust web.

A typical e-commerce transaction follows three steps: (i) locating business entities, (ii) establishing a trust path (i.e., a chain of TSPs linking the customer with the final merchant), and (iii) executing the transaction along the best-suited trust path. Successful e-commerce systems depend heavily on the second step. A trust relationship is established by an initiating entity that wishes to build a relationship with another selected entity by some means, such a private relationship, positive past experience or simply by reputation. This form of Internet-base mediation can be iteratively extended when the customer does not have a direct trust link with the final merchant [Ati02, DIM02].

To the best of our knowledge, the problem of building trust paths has been only studied from a "protocol design" perspective, by assuming that all the TSPs are *honest/obedient*, that is that they follows the protocol. In [Ati02, SM97] a distributed search algorithm has been proposed aimed at identifying a trust path linking an initiating entity e (the customer) with a set F of final merchants across a chain of TSPs such that each TSP trusts its immediate neighbors. The algorithm requires the cooperation of multiple TSPs to find

trust paths. The trust web's connectivity might be high, i.e., each TSP vertex might be linked to several other TSPs, increasing the complexity of the search process. The search space complexity can be then reduced by pruning the alternatives that do not satisfy suitable constraints, such as cost constraints (e.g., the transmission cost accumulated along the path has not to exceed a given threshold cost β).

However, the complex economic context can have a deep influence on the design process, in that if a protocol is demonstrated to have a good performance, this does not necessarily mean that it will be successful. For this protocol to be "fit", the design must be paved with *incentives* that will motivate all the participant TSPs to adopt it. Indeed, all the TSPs are actually *strategic agents* which respond to incentives and will deviate from the protocol only for a tangible gain. Mechanism design asks how one can design systems so that agents' selfish behavior results in the desired system-wide goals. That is, a mechanism should define both (i) the protocol and (ii) a suitable *payment rule* such that each agent maximizes her own utility without deviating, regardless of what other agents do. A mechanism which guarantees this property for every agent is called a *truthful mechanism*.

As regards the protocol, the model proposed in [Ati02, SM97] is based only on boolean relationships, i.e., any two entities can share either a complete trust or a complete distrust relationship. A *probabilistic trust model* based on values on a continuous scale in $[0, 1]$ has been considered in [Mau96, DIM02] for determining the most trusted path between two entities in authentication infrastructures based on public key certificates (PKI). *Authentication* is the verification of an identity of an entity, which may be performed by means of a trusted authentication service or using *certificates*[1]. There is then an issue of the degree of trust in the entity which released the certificate. Basing on this model, the degree $d_{i,j}$ of entity i's trust in entity j has been interpreted as the probability that the certificates issued by j are correct. In this paper we will adopt the probabilistic trust model and we interpret, more in general, the degree of trust in a TSP as the probability that she is capable of performing the expected functions, or the service she is meant to provide correctly and within reasonable timescales. We will incorporate such a probabilistic model in the distributed search algorithm described in [Ati02] for determining the best-suited trust paths by also pruning alternatives that does not satisfy (besides cost constraints) trust degree constraints: the "transitive" degree of trust accumulated along the path has to be greater than a given threshold value α.

More precisely, we propose an extended search algorithm (see Section 5) which allows to solve the two problems defined as follows: given a initiating entity (costumer) e, a set of final entities (merchants) F, a threshold degree of trust $\alpha \in (0, 1]$, and a threshold communication cost $\beta \in \mathbb{R}^+$, establish a set of trust paths linking e to the final entities in such a way to satisfy both cost

and trust degree constraints and then find either (i) the lowest cost trust paths reaching located merchants, or (ii) the most trusted paths reaching located merchants. We will refer to these two problems as the *min-cost-TP*$[e, F, \alpha, \beta]$ and the *max-trust-TP*$[e, F, \alpha, \beta]$ problem, respectively.

Then, we use the proposed algorithmic approach as a substrate to address the problem also from a *mechanism design* point of view. The main scope of this paper is to provide an answer to the following fundamental problem: does a payment function exist for the described problems such that the the resulting mechanism is truthful? By applying recent results appeared in [MPPW+04] and reviewed in Section 2, we provide both positive and negative answers, depending on which constraint we add/drop and on which parameters are considered as a private information of agents. In particular, we provide two distinct truthful mechanisms, called the *marginal transaction cost mechanism* and the *marginal trust mechanism*, as the solutions of two special cases of the defined problems.

2. Algorithmic mechanism design

In this section we review the basics of algorithmic mechanism design. For a more extensive discussion of applications of game theoretic tools and microeconomics to the Internet we refer the reader to [AT01, FPS01, FS02, NR99, NR00, Pap01]. In designing network protocols computer scientists typically assume that the entities involved in the computation are either *honest/obedient* (i.e., they follow the protocol) or *adversarial* (i.e., they "play against"). In contrast, game theorists study what happens when independent entities (also called *agents* or *players*) are *strategic* and respond to *incentives* (e.g., a payment received to compensate the costs). Mechanism design asks how one can design systems so that agents selfish behavior results in the desired system-wide goals.

In the standard model for the design and analysis of scenarios involving entities which act according to their own self-interest, there are n agents $\{1, \ldots, n\}$ each one holding some private information t_i, called its *type* and belonging to a type space T_i.

A *mechanism design problem* is characterized by an *output specification* $o(\cdot)$ mapping each type vector $t = (t_1, \ldots, t_n)$ to a set of feasible outputs Φ. Each agent is assumed to incur some intrinsic benefit or loss $v_i(x, t_i)$, called its *valuation*, which depends on the considered output x. A mechanism defines for each vector $r = (r_1, \ldots, r_n)$ (called the *input vector*), with $r_i \in T_i$, (i) an output $x = o(r)$ and (ii) a *payment vector* $p(r) = (p_1(r), \ldots, p_n(r))$. Each agent i gives $r_i \in T_i$ as input (i.e., agent i plays r_i) in order to maximize her own *utility* $u_i(o(r), t_i, p_i(r))$, expressible as a function of the output $x = o(r)$, the valuation $v_i(x, t_i)$ and the payment $p_i(r)$. Therefore, one should design both (i) an algorithm $\mathcal{A}$ which computes $o(\cdot)$ and (ii) a suitable payment rule

$p(\cdot)$ such that each agent maximizes her utility by "playing" the type $r_i = t_i$ regardless of what other agents do. In other words, if r^{-i} denotes a vector of inputs given by all agents except agent i, the relation

$$u_i(o(r^{-i}, t_i), t_i, p_i(r^{-i}, t_i)) \geq u_i(o(r^{-i}, r_i), t_i, p_i(r^{-i}, r_i)) \qquad (1)$$

must hold for all i and all possible values of t_i, r^{-i} and r_i.

The pair $(\mathcal{A}, p)$ that allows to guarantee Property (1) for every agent is called a *truthful mechanism* with dominant strategies. Hence, a mechanism wants each agent to report her type truthfully, and it is allowed to pay agents in order to provide incentives for them to do so.

A large body of the existing literature focuses on the class of problems in which the utilities are *quasi-linear*, that is, agent i's utility factors into $u_i(o(r), t_i, p_i(r)) = v_i(o(r), t_i) + p_i(r)$. For such problems, the celebrated Vickrey-Clarke-Groves (VCG) mechanisms [Cla71, Gro73, Vic61] guarantee the truthfulness under the hypothesis that the algorithm maximizes the objective function $\mu(x, r) = \sum_i v_i(x, r_i)$. VCG mechanisms have been successfully applied to a multitude of optimization problems involving selfish agents with applications to networking [FPS01, NR00] and e-commerce [Cra97]. All these works assume that the problem is *utilitarian*, that is, the utility functions are quasi-linear and the objective function can be written as the sum above.

Recently, a class of optimization problems has been defined in [MPPW+04], termed *consistent problems*, which are mechanism design problems such that:

(a) the set of feasible solutions Φ does not depend on agents' types;

(b) the *consistent objective function* is expressible as $\mu(x, r) = \bigoplus_i v_i(x, r_i)$ for a suitable operator $\oplus$ which enjoys the following properties: associativity, commutativity and monotonicity in its arguments;

(c) the utility function is expressible as $u_i(o(r), t_i, p_i(r)) = v_i(o(r), t_i) \oplus p_i(r)$.

The authors proved that consistent problems admit truthful mechanisms, called *VCG-consistent* (VCGc) mechanisms and defined as a natural extension of VCG mechanisms. If the operator also enjoys the following properties: identity element, inverse and strict monotonicity, the VCGc mechanisms are the only truthful mechanisms for the problem. The characterization of VCGc mechanisms states that the payment functions $p_i(\cdot)$ provided from a truthful mechanism for a consistent problem can be expressed as $p_i(r) = \bigoplus_{j \neq i} v_j(o(r), r_j) \oplus h_i(r^{-i})$, where $h_i(\cdot)$ is a function of r^{-i}. Interestingly, it has been also identified a subclass of non-consistent problems in which the set of feasible solutions Φ depends on agents' types which does not admit truthful mechanism.

3. The model

We consider a graph-theoretic model for the *trust web* referred to here as the *trust graph* and composed of:

- a set $V = \{1, \ldots, n\}$ of weighted vertices (the agents) denoting each a TSP whose weight $c_i \in \mathbb{R}^+$ represents the cost TSP i incurs for executing each transaction;
- a set E of weighted directed edges $\langle i, j \rangle \in V \times V$, which reflect trust relationships, whose weight $d_{i,j} \in (0, 1]$ denotes the degree of trust of i in j, according to the probabilistic model proposed in [Mau96, DIM02].

From now on we will adopt the following notations. Let $\mathcal{E}$ be any set of potential transacting entities (e.g., consumers and merchants) such that $V \cap \mathcal{E} = \emptyset$. We will refer to e as the entity in $\mathcal{E}$ initiating a request for carrying out an e-commerce transaction (said a consumer) to acquire commodities from a set of final entities f (merchants). For any TSP $i \in V$, $\mathcal{E}[i] \subseteq \mathcal{E}$ denotes all the entities that know and trust TSP i directly, and, for any $x \in \mathcal{E}$, $V[x] \subseteq V$ is the subset of TSPs known and trusted by x, i.e., $V[x] = \{i \in V \mid x \in \mathcal{E}(i)\}$. Finally, for each TSP i, we will denote by $N(i)$ the set of neighbors of i in the trust graph.

DEFINITION 1 *A* trust path $\pi(e, f)$ *linking* e *to* f *is a chain of TSPs* $\langle i_1, \ldots, i_k \rangle$ *such that* $e \in \mathcal{E}(i_1)$, $f \in \mathcal{E}(i_k)$, *and for each* $j = 2, \ldots, k$, *it holds* $i_j \in N(i_{j-1})$.

DEFINITION 2 *The* transaction cost of a trust path π, *denoted by* $cost(\pi)$, *is the overall cost incurred by each TSP* i *on the path, i.e.,* $cost(\pi) = \sum_{i \in \pi} c_i$.

DEFINITION 3 *The* degree of a trust path $\pi = \langle i_1, \ldots, i_k \rangle$, *denoted by* $d(\pi)$, *is the "transitive" degree of trust induced by* π, *i.e.,* $d(\pi) = \prod_{j=1}^{k-1} d_{i_j, i_{j+1}}$.

Let $\Pi(e, f)$ be the set of all the different trust paths in the trust graph linking e to f.

DEFINITION 4 (THE LOWEST-COST TRUST PATH) *The* lowest-cost trust path $\pi_{LC}(e, f)$ *linking* e *to* f *is the trust path in* $\Pi(e, f)$ *of minimum transaction cost.*

DEFINITION 5 (THE MOST-TRUSTED PATH) *The* most-trusted path $\pi_{MT}(e, f)$ *linking* e *to* f *is the trust path in* $\Pi(e, f)$ *of maximum transitive trust degree.*

4. The problems

A typical e-commerce transaction for acquiring commodities from a set of merchants F (final entities or targets) consists of the following steps:

- Locating a set $R \subseteq F$ of reachable final merchants.
  Let ρ denote the characteristic function associated with R, i.e., $\rho(f) = 1$ for every $f \in R$, $\rho(f) = 0$ otherwise.
- Constructing a set of trust paths $\Pi(e, R, \alpha, \beta)$ linking e to finals in R, where α e β are two parameters whose meaning will be clear later.
  We denote by $\Pi(e. f, \alpha, \beta) \subseteq \Pi(e, R. \alpha, \beta)$ the subset of trust paths linking e to f.
- For any final $f \in R$, executing the transaction along either (i) a lowest-cost trust path $\pi_{\mathrm{LC}}(e, f) \in \Pi(e, f, \alpha, \beta)$ (see Def. 6), or (ii) a most-trusted path $\pi_{\mathrm{MT}}(e, f) \in \Pi(e, f, \alpha, \beta)$ (see Def. 7).
  Let χ_{LC} and χ_{MT} be the characteristic functions for the lowest-cost trust path and the most-trusted path, respectively, i.e., $\chi_{\mathrm{LC}}(i, e, f) = 1$ if $i \in V$ is on the path $\pi_{\mathrm{LC}}(e, f)$ and $\chi_{\mathrm{LC}}(i, e, f) = 0$ otherwise, and $\chi_{\mathrm{MT}}(i, e, f) = 1$ if $i \in V$ is on the path $\pi_{\mathrm{MT}}(e, f)$ and $\chi_{\mathrm{MT}}(i, e, f) = 0$ otherwise.

Minimizing for every $f \in R$ the cost of the trust path linking e to f is equivalent to minimizing the overall transaction cost. Hence:

DEFINITION 6 *For every feasible output $x = (\rho, \chi_{\mathrm{LC}})$, the* min-cost-TP$[e, F, \alpha, \beta]$ problem *aims to minimize the following objective function:*

$$\mu_{\mathrm{LC}}(x, e, F, \alpha, \beta) = \sum_{f \in F} \rho(f) \sum_{i \in V} c_i \cdot \chi_{\mathrm{LC}}(i, e, f). \tag{2}$$

Equivalently:

DEFINITION 7 *For every feasible output $x = (\rho, \chi_{\mathrm{MT}})$, the* max-trust-TP$[e, F, \alpha, \beta]$ problem *aims to maximize the following objective function:*

$$\mu_{\mathrm{MT}}(x, e, F, \alpha, \beta) = \prod_{f \in F} \rho(f) \prod_{i \in V} \chi_{\mathrm{MT}}(i, e, f) \left(\sum_{\langle i,j \rangle \in E} \chi_{\mathrm{MT}}(j, e, f) \cdot d_{i,j} \right). \tag{3}$$

5. The trust paths building algorithm

In this section we describe a search algorithm $\mathcal{A}$ which solves the just introduced problems. In order to identifying the best-suited trust paths, as each TSP within the trust web knows only its immediate trusted neighbors (each $i \in V$ only knows adjacent vertices $N(i)$ within the trust graph) rather than each TSP within the global trust web, the algorithm $\mathcal{A}$ requires the cooperation of multiple TPSs.

When a customer e initiates the search process, it relies the final merchants F automatically to the nearest trusted TSPs $V[e]$ in hopes of finding the finals. If these TSPs do not have direct trust relationships with the finals, they *forward* the customer's request to the adjacent TSPs to identify finals deeper within the trust graph. Therefore, the search of finals generates *forward messages*. A TSP stops the search when there are no connections or all its connections lead to already explored TSPs[2], or when all the connections lead to "unreliable" ot "too expensive" trust paths. Indeed, the search space complexity and the number of messages exchanged between TSPs is reduced by pruning the alternatives π that does not satisfy these constraints:

- *Trust constraint*: the trust degree accumulated along the path is greater than a threshold trust degree $\alpha \in [0,1]$, i.e., $d(\pi) = \prod_{\langle i,j \rangle \in \pi} \geq \alpha$;
- *Cost constraint*: the transaction cost accumulated along the path is less than a threshold cost $\beta \in I\!R^+$, i.e, $cost(\pi) = \sum_{i \in \pi} c_i \leq \beta$.

When finals have been located, *backward messages* trace the path of the forward messages back to e. If e does not receive a backward message after a certain time, it assumes that no path was found.

The structure of a *forward message* $m = \{e, F, \alpha, \beta, \pi_r, d(\pi_r), cost(\pi_r)\}$ specifies (among the others):

- the costumer e and the list of final merchants F to be located;
- a threshold trust degree $\alpha \in [0,1]$ and a threshold transaction cost $\beta \in I\!R^+$;
- a *return path* π_r containing a sequence initially empty of 4-tuples

$$\langle (i_1, F_{i_1}, d_{e,i_1} = 1, c_{i_1}), \ldots, (i_h, F_{i_h}, d_{i_{h-1},i_h}, c_{i_h}) \rangle$$

  where $\langle i_1, \ldots i_h \rangle$ is the identified path $\pi(e, i_h)$ and, for any TSP i_j on it, F_{i_j} is the set of identified targets that i_j knows and trusts directly;
- the trust degree $d(\pi_r) = d(\pi(e, i_h))$ accumulated along the path $\pi(e, i_h)$;
- the transaction cost $cost(\pi_r) = cost(\pi(e, i_h))$ accumulated along the path $\pi(e, i_h)$.

For a message $m = \{e, F, \alpha, \beta, \pi_r, d(\pi_r), cost(\pi_r)\}$ identifying a trust path $\pi(e, i)$, we say that a TSP j is *appended* to m if:

- the 4-tuple $(j, F \cap \mathcal{E}[j], d_{i,j}, c_j)$ is added to the return path π_r;
- F is updated to the set of TSPs left to locate, i.e., $F = F \setminus \mathcal{E}[j]$;
- the trust degree is updated to $d(\pi_r) \cdot d_{i,j}$;
- the transaction cost is updated to $cost(\pi_r) + c_j$.

The resulting message is denoted by $append(m, i, j)$.

The algorithm works as follows (due to lack of space details concerning the backward phase will be omitted):

- Initially, if customer e neither knows nor trusts finals in F, it prepares the initial forward message $m = \{e, F, \alpha, \beta, \pi_r = \emptyset, d(\pi_r) = 1, cost(\pi_r) = 0\}$ and sends it in parallel to all trusted TSPs $j \in V[e]$. Without loss of generality, we assume that consumers/merchants completely trust TSPs they know directly (i.e., $d_{e,j} = 1, \forall j \in V[e]$).
- On receiving a message $m = \{e, F, \alpha, \beta, \pi_r, d(\pi_r), cost(\pi_r)\}$ from i a TSP j executes the following steps:
  - let $N(j)' = \{k \in N(j) \mid k \text{ is unvisited} \wedge d(\pi_r) \cdot d_{j,k} \geq \alpha\}$;
  - if $cost(\pi_r) + c_j > \beta \vee N(j)' = \emptyset$ (this is a dead end) then
    if $F \cap \mathcal{E}[j] = \emptyset$ then j sends m backward to i
    else j sends $append(m, i, j)$ backward to i;
    else j forwards $append(m, i, j)$ to each $k \in N(j)'$.

6. The mechanism design trust paths building problems

The algorithm just described assumes that that all the reached TSPs are *honest/obedient*, that is that they follows the protocol when reporting their transaction costs and the degrees of trust in the neighbors, which are both a private information of TSPs (their types). However, all the TSPs are actually *strategic agents* which respond to incentives and may deviate from the protocol for a tangible gain. Hence, for this algorithm to be "fit", the design must be paved with suitable *payments* that will motivate all the participant TSPs to adopt it.

In this section we aim to answer the following central question about the study of which goals are achievable via truthful mechanisms, that is "does a payment function $p(\cdot)$ exist such that the resulting mechanism is truthful?"

Towards this aim, we need to define the trust path building problem as a consistent problem which we know to admit truthful mechanisms. It is worthy to notice that we provide a payment specified by $p_i(\cdot)$ to a TSP i if and only if i is on one of the established trust paths linking the consumer e to some merchant in F.

Since e-commerce transactions are executed along either the most trusted paths or the lowest cost trust paths, the definition of *consistent problem* (requiring that the set of feasible solutions does not depend on agents' types) points to only two distinct mechanisms, the *marginal transaction cost mechanism* and the *marginal trust mechanism* as the two solutions of the two mechanism design problems defined in Sections 6 and 6.

The min-cost-TP[e, F, α] problem

Given the set $V = \{1, \ldots, n\}$ of agents (TSPs within the trust web), formulating the *min-cost-TP*$[e, F, \alpha, \beta]$ problem as a mechanism design consistent problem basically requires (i) to define agent's types and prove that the

set of feasible solutions Φ is independent of them; (ii) to define the valuation functions $v_i(\cdot)$ and show that the objective function is of the form $\bigoplus_{i=1}^{n} v_i(\cdot)$, where $\oplus$ is a suitable operator which enjoys the following properties: associativity, commutativity and monotonicity in its arguments; (iv) to express in a "consistent" way the utility functions as $u_i(\cdot) = p_i(\cdot) \oplus v_i(\cdot)$ for appropriate payment functions $p_i(\cdot)$.

The scenario described until now considers each TSP i as a strategic agent which knows both the transaction cost c_i and the degrees of trust. However, by following this model the set of feasible solutions would completely depend on agents' types. These dependencies would affect the existence of truthful mechanisms for the problem [MPPW+04]. Hence, the only way to structure the problem as a consistent problem is to make feasible solutions $x = (\rho, \chi_{\mathrm{LC}})$ independent of transaction costs by assuming a modified scenario where:

- degrees of trust are publicly known and agent i's type is the transaction cost $t_i = c_i$, for each $i \in V$;
- the search space is not reduced by applying cost constraints.

We denote such a subproblem by *min-cost-TP*$[e, F, \alpha]$.

It is immediate to verify that the function (2) is a *utilitarian* objective function (i.e., a consistent objective function whose operator is the sum) by defining the valuation functions as follows: for any reported vector of costs r and for any feasible output $x = (\rho, \chi_{\mathrm{LC}})$, i.e., a set of located finals $R \subseteq F$ and a set of trust paths each linking e to a final entity $f \in R$, we have

$$v_i(x, r_i) = \sum_{f \in F} \rho(f) \cdot r_i \cdot \chi_{\mathrm{LC}}(i, e, f). \tag{4}$$

THEOREM 8 *When the distributed algorithm $\mathcal{A}$ outputs a solution $x^* = (\rho, \chi_{\mathrm{LC}})$ inducing on the trust graph G a biconnected subgraph and picks lowest cost paths linking e to the reachable finals, then there is a unique truthful mechanism $(\mathcal{A}, p)$ for the* min-cost-TP$[e, F, \alpha]$ *problem that gives no payment to TSPs not involved in transactions.*

Proof. For any reported vector of costs r and for any feasible solution x, let us denote by $\mu_{\mathrm{LC}}(x, r) = \sum_{i=1}^{n} v_i(x, r_i)$ the overall transition cost of the solution. Moreover, let $\mu_{\mathrm{LC}}^{-i}(x, r) = \sum_{j \neq i}^{n} v_j(x, r_j)$. For utilitarian problems, VCG mechanisms [Cla71, Gro73, Vic61] guarantee the truthfulness of the mechanism $(\mathcal{A}, p)$ when the payments to the involved TSPs i are of the form: $p_i(r) = \mu_{\mathrm{LC}}(\mathcal{A}(r^{-i}, r_i = \infty), r) - \mu_{\mathrm{LC}}^{-i}(\mathcal{A}(r), r)$. □

The max-trust-TP$[e, F, \beta]$ problem

Similarly to the *min-cost-TP*$[e, F, \alpha, \beta]$ problem, in order to formulate the *max-trust-TP*$[e, F, \alpha, \beta]$ problem as a mechanism design consistent problem

we need to make feasible outputs $x = (\rho, \chi_{\mathrm{MT}})$ independent of degrees of trust by assuming a modified scenario where:

- costs are publicly known and agent i's type is the vector t_i of degrees of trust $t_i^j = d_{i,j}$, for each $j \in N(i)$;
- the search space is not reduced by applying trust constraints.

We denote such a subproblem by *max-trust-TP*$[e, F, \beta]$.

In order to prove that the function (3) is a consistent objective function whose operator is the standard product, we define the valuation functions as follows. For any reported degrees of trust r and for any feasible output $x = (\rho, \chi_{\mathrm{MT}})$:

$$v_i(x, r_i) = \prod_{f \in F} v_i^f(x, r_i) \quad \text{such that}$$

$$v_i^f(x, r_i) = \begin{cases} \sum_{\langle i,j \rangle \in E} r_i^j \cdot \chi_{\mathrm{MT}}(j, e, f) & \text{if } \rho(f) + \chi_{\mathrm{MT}}(i, e, f) = 2; \\ 1 & \text{otherwise.} \end{cases}$$

THEOREM 9 *When the distributed algorithm $\mathcal{A}$ outputs a solution $x^* = (\rho, \chi_{\mathrm{MT}})$ inducing on the trust graph G a biconnected subgraph and picks most trusted paths linking e to the reachable finals, then there exists a truthful mechanism $(\mathcal{A}, p)$ for the* max-trust-TP$[e, F, \beta]$ *problem that gives no payment to TSPs not involved in transactions.*

Proof. For any feasible solution x and for any reported vector of degrees of trust r, let $\mu_{\mathrm{MT}}(x, r) = \prod_{i=1}^{n} v_i(x, r_i)$ be the overall trust degree of the solution and let $\mu_{\mathrm{MT}}^{-i}(x, r) = \prod_{j \neq i}^{n} v_j(x, r_j)$. For consistent problems, the VCGc mechanisms recently introduced in [MPPW+04] guarantee the truthfulness of the mechanism $(\mathcal{A}, p)$ when the payments to the involved TSPs i are of the form:

$$p_i(r) = \frac{\mu_{\mathrm{MT}}^{-i}(\mathcal{A}(r), r)}{\mu_{\mathrm{MT}}(\mathcal{A}(r^{-i}, r_i = 0), r)}.$$

□

Acknowledgments

This work has been developed within the activities of the Research Project GRID.IT, funded by the Italian Ministry of Education, University and Research, and whose support is gratefully acknowledged. The authors would also like to thank Enrico Nardelli for useful discussions on the topic presented here.

Notes

1. A certificate is a digitally signed statement by which a trusted third party, referred to as a *Certification Authority* (CA), asserts that a public key is bound to an entity. The term PKI is used to refer to an infrastructure for distributing public keys, where the authenticity of public keys is certified by the CAs.

2. To avoid a TSP receive and process the same request twice we suppose that (i) each request includes a unique session ID and time-stamps, (ii) all received requests are backlogged and (iii) duplicated requests are discarded.

References

[AT01] A. Archer and E. Tardos. Truthful mechanisms for one-parameter agents. In *Proc. of the 42nd Annual Symposium on Foundations of Computer Science (FOCS'01)*, 482–491, 2001.

[Ati02] Y. Atif. Building Trust in E-Commerce. *IEEE Internet Computing*, 6(1):18–24, 2002.

[CAR00] G. Caronni. Walking the web of trust. In *Proc. of 9th Workshop on Enabling Technologies (WETICE 2000)*, IEEE Computer Society Press, 153–158, 2000.

[Cla71] E. Clarke. Multipart pricing of public goods. *Public Choice*, 8:17–33, 1971.

[Col90] J. Coleman. *Foundations of Social Theory*. Harvard University Press, 1990.

[Cra97] P. Cramton. The FCC spectrum auction: an early assessment. *Journal of Economics and Management Strategy*, 6:431–495, 1997.

[FPS01] J. Feigenbaum, C.H. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63(1):21–41, 2001.

[FS02] J. Feigenbaum and S. Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In *Proc. of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, ACM Press, 1–13, 2002.

[GL77] J. Green and J.J. Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45(2):727–738, 1977.

[Gro73] T. Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973.

[HFH99] B.A. Hubermann, M. Franklin, and T. Hogg. Enhancing privacy and trust in electronic communities. In *Proc. of the 1st ACM Conf. on Electronic Commerce (EC'99)*, 78–86, 1999.

[Mau96] U. Maurer. Modelling a Public-Key Infrastructure. In *Proc. of the 4th European Symposium on Research in Computer Security (ESORICS'96)*, Lecture Notes in Computer Science vol. 1146, Springer-Verlag, 325–350, 1996.

[DIM02] F. Di Vito, P. Inverardi, and G. Melideo. A context-aware approach to infer trust in Public Key Infrastructures. In *Proc. of the 17th IFIP World Computer Congress - International Workshop on Certification and Security in E-Services (CSES 2002)*, Kluwer Academic Publishers, 111–125, 2002.

[MPPW+04] G. Melideo, P. Penna, G. Proietti, R. Wattenhofer, and P. Widmayer. Truthful mechanisms for generalized utilitarian problems. In *3rd IFIP International Conference on Theoretical Computer Science (TCS'04)*, August 24-26, 2004, Toulouse, France. Proceedings will be published by Kluwer.

[NR99] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behaviour*, 35:166–196, 2001.

[NR00] N. Nisan and A. Ronen. Computationally feasible VCG mechanisms. In *Proc. of the 2nd ACM Conference on Electronic Commerce (EC 2000)*, 242–252, 2000.

[Pap01] C. H. Papadimitriou. Algorithms, games, and the Internet. In *Proc. of the 33rd Annual ACM Symposium on Theory of Computing (STOC'01)*, 749–753, 2001.

[SM97] J. Su and D. Manchala. Building trust for distributed commerce transactions. In *Proc. of the 17th Int. Conf. Distributed Computing Systems*, IEEE CS Press, 322–329, 1997.

[Vic61] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.