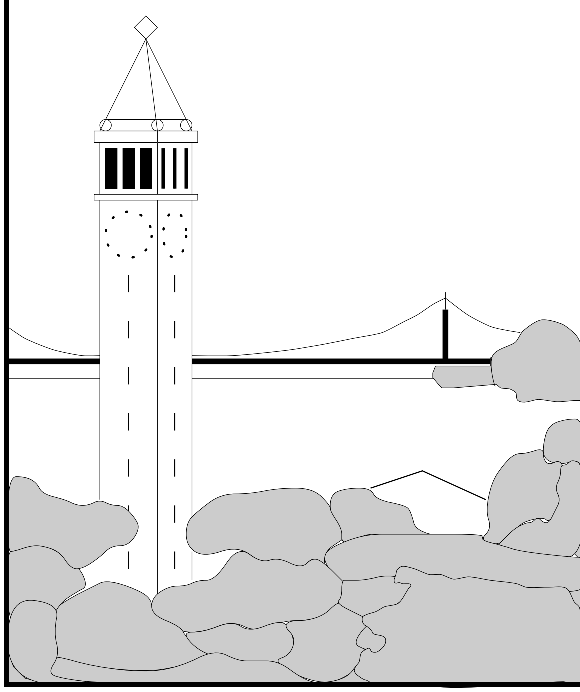
Reconciling Cooperation with Confidentiality in Multi-Provider Distributed Systems

Sridhar Machiraju and Randy H. Katz
EECS Department, University of California, Berkeley



Report No. UCB/CSD-4-1345

August 2004

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Reconciling Cooperation with Confidentiality in Multi-Provider Distributed Systems

Sridhar Machiraju, Randy H. Katz
EECS Department, University of California, Berkeley

Technical Report UCB//CSD-04-1345

Cooperation and competition are opposing forces in Multi-Provider Distributed Systems (MPDSs) such as the Internet routing infrastructure. Often, competitive needs cause providers to keep certain information confidential thereby hindering cooperation and leading to undesirable behavior. For instance, recent work has shown that lack of inter-domain cooperation in performing intra-domain routing changes may cause more congestion. We argue that MPDSs should be designed with mechanisms that enable cooperation without violating confidentiality requirements. We illustrate this design principle by developing such mechanisms to solve well-known problems in the most successful MPDS, inter-domain routing. We also briefly discuss the need for such mechanisms in MPDSs for content distribution and policy-based resource allocation. Our mechanisms leverage secure multi-party computation primitives.

1 Introduction

Since the Internet was commercialized in 1994-1995, its routing infrastructure has evolved into a *multi-provider distributed system (MPDS)*. The protocol used to exchange routing information in this MPDS is BGP[33] (Border Gateway Protocol), a hop-by-hop policy-aware path vector protocol. For scalability reasons, BGP hides details about a provider's network (called an AS, short for Autonomous System) as much as possible[12]. Over the years, this property has allowed AS operators to preserve the confidentiality of their policies, topology, operational status and other intra-domain information.

The opaqueness of BGP is the cause of many problems. For instance, lack of knowledge about congestion in neighboring ASs makes it hard for one AS to choose uncongested end-to-end paths [43, 14]. Worse still, conflicting AS policies can cause BGP to diverge [41]. Also, path quality is not available for ASs to use in path selection [6, 27]. Since most intra-domain information is considered confidential, the above and similar problems cannot be solved by requiring ASs to reveal their intra-domain information. In this paper, we develop techniques that show how confidential information such as link qualities and policies may be shared. We also briefly discuss the potential use of such techniques in MPDSs other than inter-domain routing such as caching [5], computing [16, 18], storage [25] and p2p networks/overlay routing[35].

The goal in the area of *Distributed Algorithmic Mechanism Design* (DAMD) is to remove the rationale for keeping certain inputs private by designing mechanisms in which it can be proven that lying does a participant no good. Our work is based on the belief that this assumption is not appropriate for MPDSs. To quote Feigenbaum et al. [15] on the applicability of DAMD to inter-domain routing, *The premise of the (DAMD) approach is that agents will voluntarily reveal their private information if it can be proven that lying does them no good in the situation (being) addressed . . . Revelation of private information may be an agent's best possible strategy for the particular situation at hand but it may be unacceptable in the broader context.* As they further state in their paper, the goal is to compute a global optimum based on certain private inputs, which is what techniques developed for *Secure Multi-Party Computations* (SMPC, in short; see [1] for a list of important papers) do. Indeed, many of our proposed techniques borrow ideas developed by the cryptography community for SMPC. However, we use techniques to solve specific problems whereas the traditional emphasis has been on developing a generic way to evaluate any function securely.

We use BGP [1] to illustrate how cooperation and confidentiality can be reconciled in MPDSs. Two of our protocols enable an AS to perform traffic engineering while considering network conditions in a neighboring AS. We also develop a protocol that shows how confidential AS policies may be used in algorithms to detect potential routing divergence. Apart from extending previous work [10] on solving a linear programming (LP) problem, we also use homomorphic encryption schemes and commutative encryption schemes. Our designs reflect two key characteristics of MPDSs, the presence of an out-of-band relationship which can be leveraged to prevent certain kinds of malicious behavior such as abnormal protocol termination, and the need to consider information leakage over multiple instances of the protocol.

The organization of this paper is as follows. In Section 2, we provide a brief overview of BGP. We then state our goals, assumptions and threat model using BGP as an illustrative MPDS. In Section 3, we discuss basic cryptographic primitives that we use in our protocols. In Section 4, we consider problems in BGP that occur due to lack of information on operational conditions in ASs and develop protocols that solve these problems without violating confidentiality. In Section 5, we show the need for sharing policy information and develop protocols that enable this without violating confidentiality requirements. In Section 6, we discuss other potential applications of SMPC to enable sharing of information on operational conditions and policies in inter-domain routing, CDNs and policy-based resource allocation. We present some preliminary results in these areas and discuss open questions in these areas. We conclude in Section 7. For ease of exposition, we do not use a separate section describing related work; Instead, we mention it whenever necessary.

2 Overview of MPDSs

In this section, we provide a brief overview of BGP and identify information that is considered confidential by network operators. Using inter-domain routing as an example MPDS, we specify our goals, assumptions and threat model in MPDSs.

2.1 Inter-domain Routing and BGP

The Internet consists of multiple Autonomous Systems (ASs) which use an inter-domain routing protocol, Border Gateway Protocol (BGP) [33], to route packets across ASs. Reachability information is exchanged on a per-prefix (not per-address) basis. Each AS advertises destination prefixes that belong within it to its neighbors. ASs also propagate routes advertised by a neighbor to other neighbors. ASs use *BGP path selection rules* to select from the various routes advertised to a destination prefix. BGP is also a policy-aware protocol because each step of the protocol can be modified by policies set by the network operators. For instance, policy could be used to decide which advertisements are preferred. Adjacent ASs are said to peer with each other. They may do so at multiple peering locations. Such BGP peering relationships also involve various agreements that indicate the amount of traffic that may be exchanged at peering points, the cost of sending/receiving traffic etc.

A large spectrum of intra-domain information is considered confidential by network operators because of commercial reasons and also for purposes of preventing attacks on the weaker parts of the network. Efforts to deduce intra-domain information of ASs such as topology [37], link characteristics [26],operational conditions [4], policies [36], AS relationships [38] and BGP configuration [42] have achieved varying degrees of success. Nevertheless, most intra-domain information is considered confidential.

2.2 Goals, Assumptions and Threat Model

Our goal in this paper is to explore instances of undesirable behavior in MPDSs on account of unavailability of confidential information of individual providers and to develop mechanisms that would enable such confidential information to be used. To understand how such undesirable behavior may occur, we use Figure 1. Flow f can exit A at either of the two peering points. Considering the available bandwidth of the path to these two peering points in A, peering point 1 is better. However, if A had knowledge of routes in B and available bandwidths on them, the

[1] For brevity, we refer to BGP and "inter-domain routing" interchangeably.

conclusion would be exactly opposite. In this case, internal topology and operational conditions of B are hidden from A and hence, A chooses the wrong route.

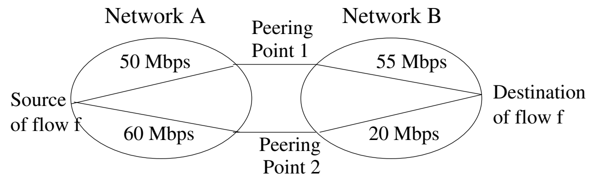


Figure 1: Peering point 2 provides the better route in A whereas Peering point 1 provides the better overall route, considering available bandwidth in A and B.

The outcome of a secure mechanism could reveal some information on the private inputs of other providers. Hence, as is typically done when evaluating SMPC, we deem a technique successful if it does not leak any more information than is deducible from the outcome. This goal must be qualified, though. A decision to use a successful technique in MPDSs must also consider information leakage from multiple invocations of the technique too. For instance, a single comparison operation may only indicate the possible range of a number; However, the exact value of a number can easily be deduced in $O(log(n))$ comparison operations.

The most important characteristic of the threat model of MPDSs is the existence of an out-of-band relationship (e.g., BGP peering agreements). Economic penalties specified in these relationships can be used to prevent misbehavior such as abnormal protocol termination. Hence, *unfairness* (one participant knowing more than the other) is not a concern in MPDSs. Failures should not be classified as abnormal terminations. Hence, participants using our protocols need to save all relevant state until it is completed so that a protocol that terminates due to failures can be continued afterwards.

The out-of-band relationship is similar, in principle, to the third party in the optimistic model for secure computation [8]. In cases that require more than 2 participants, we assume the presence of a connected graph of bilateral relationships. We assume that this can be achieved using protocols outside of BGP or in architectures similar to those discussed in [2] and [20].

We assume secure channels between the participants of an MPDS, i.e., no adversary can snoop, inject or modify traffic sent by the participants in the MPDS. We only analyze possible information leakage from using our techniques. Hence, adversarial behavior is restricted to participants in the protocol. Adversaries are assumed to be able to collude and provide inconsistent inputs only under three conditions. Such misbehavior should allow the adversaries to determine the inputs of honest participants, improve their own system or degrade the system of honest participants. Finally, malicious behavior that has a high probability of detection can be discounted since the out-of-band relationship can be used to impose penalties.

3 Cryptographic Primitives

In this section, we briefly describe related work in cryptographic literature relevant to our solutions including SMPC. One commonly used property of cryptosystems in SMPC is the homomorphic property, i.e., if E represents the encryption operation, then $E(m_1)E(m_2) = E(m_1 * m_2)$ where $*$ is either the addition or multiplication operation. In the case of the former, E is said to possess additive homomorphism and in the case of the latter, E is said to possess multiplicative homomorphism. For instance, El Gamal[11] and Paillier's [29] are multiplicative and additive respectively. Appendix A.1 provides a brief overview of these cryptosystems. Unless stated otherwise, we use additive homomorphic cryptosystems in this paper. Below, we state important properties of these cryptosystems useful to us:

- $E(m_1) \cdot E(m_2) = E(m_1 + m_2)$, the additive homomorphic property. Also, $E(m_1) \cdot f(m_2) = E(m_1 + m_2)$ where $f()$ has much lesser complexity than E. With Paillier's, $f(m_2) = g^{m_2}$.

- $E(m)^k = E(mk)\ \forall\ k$. A special case is the calculation of $E(-m)$.
- The ciphertexts before and after performing one or more of the above operations can be used to deduce the operation. For instance, upon calculating $E(-m)$, $E(m) = E(-m)^{-1}$. In such cases, $E(-m)$ can be *blinded* so that the operation cannot be deduced. With Paillier's, this can be done by multiplying $E(-m)$ with r'^n where r' is a random number. Note that such blinding factors can be precomputed.

Threshold cryptography refers to n entities sharing a decryption key. With these schemes, encryption can be done with knowledge of the public key while decryption requires contributions from at least t of these entities. Threshold variants have been proposed for many cryptosystems. For instance, [17, 9] are threshold variants of Paillier's cryptosystem. Note that, the shares of the decryption key may be computed by a third party or by the participants in a distributed manner. Unless specified otherwise, we assume that $t = n$ in threshold cryptosystems, i.e., all shares of a secret key are required for the decryption operation. We describe cryptographic primitives that we use in Appendix A.

4 Sharing Operational Conditions in BGP

Lack of knowledge on operational conditions within other ASs makes intra-domain route selection hard. In this section, we develop two protocols that enable neighboring ASs to share information about internal congestion conditions which can be used to determine how traffic is exchanged between them. Though the scenarios we consider are motivated by those considered in BGP-related literature, our protocols are broad enough to be useful in any inter-domain routing system (including overlay networks).

4.1 Safe Traffic Engineering

As pointed out in [43, 14], coordinating traffic engineering with neighboring ASs is desirable due to cases similar to those shown in Figure 1. To develop techniques for enabling such cooperation, we consider the problem first posed in [43]. AS A and B peer with each other at multiple peering locations. For various reasons, the operator of A wants to change her intra-domain routes. This affects the traffic matrix from A to B. For instance, traffic to a destination could ingress B at a different peering point than before. The goal is for the operator of A to make traffic matrix changes such that the quality of the new routes through B is good. The two naive ways of doing this are either for A to know Bs topology and link characteristics or for B to know link characteristics in A. Both of these are undesirable since these require an AS to reveal confidential information. The best solution known [43] lets B get some knowledge on proposed changes to the traffic patterns from A and A some knowledge of Bs topology and does not consider adversarial behavior.

We now develop a technique that allows A to determine if a traffic matrix change is acceptable to B or not, i.e., does not saturate links in B, without violating the confidentiality requirements of the ASs. Table 1 lists the various variables we use to explain our technique. A's goal is to determine if the traffic matrix change, $\vec{\Delta}$, will saturate no link in B. Define s_l to be the difference between available bandwidth and requested bandwidth on link e_l. The following must be true for $\vec{\Delta}$ to be accepted.

$$\vec{B} \geq \sum_{(i,j)} \delta_{i,j} \vec{U}_{i,j} \tag{1}$$

$$\implies s_l = b_l - \sum_{(i,j)} \delta_{i,j} u_{i,j,l} \geq 0\ \forall l. \tag{2}$$

The following protocol implements the above computation securely. A is assumed to use an asymmetric key pair; E_A, D_A denote the encryption, decryption operations with this pair. The complexity of the protocol is a function of the total number of destinations, D, the number of peering points between the two ASs, P and the total number of bottleneck links in B, N.

Protocol Description:

Table 1: Notation used for Safe Traffic Engineering Scenarios. The second half of the table is used only in Section 4.2

Variable	Description	Private to
$\{d_i\} 1 \le i \le D$	Unique destination prefixes	None
$\{p_j\} 1 \le j \le P$	Peering points between A and B	None
$\{e_l\} 1 \le l \le N$	Potential bottleneck links in B	B
$u_{i,j,l}$; $\vec{U}_{i,j} = (\{u_{i,j,l}\}_l)$	1 if route to dest. d_i from p_j uses link e_l, 0 otherwise	B
b_l; $\vec{B} = (b_1, \ldots b_N)$	available bandwidth on link e_l of B	B
$\delta_{i,j}$; $\vec{\Delta} = (\{\delta_{i,j}\}_{i,j})$	Proposed change in traffic to d_i entering B at p_j	A
$\{f_h\} 1 \le h \le F$	"Heavy-hitter" flows	None
c_h	Capacity of flow f_h	None
x_{hj}	Amount of flow f_h through p_j	None
$\{e_k\} 1 \le k \le M$	Potential bottleneck links in A	A
a_k; $\vec{A} = (a_1, \ldots a_M)$	available bandwidth on link e_k of A	A
$\alpha_{hjk} \forall h, j, k$	1 if link e_k is used when f_h exits A at p_j	A
$\beta_{hjl} \forall h, j, l$	1 if link e_l is used when f_h exits B at p_j	A

- A calculates $E_A(-\delta_{i,j}) \; \forall \; (i, j)$ and sends them to B. This requires $O(DP)$ encryptions and causes communication overhead of $O(DP)$.

- For each link e_l, B calculates $E_A(s_l)$. It can do this by computing $E_A(b_l)$, $E_A(-\delta_{i,j})$ and the homomorphic property. The computation complexity is $O(N)$ encryptions and $O(DPR)$ multiplications where R is the average number of non-zero $u_{i,j,l}$ values of a link e_l. This is equal to the average number of bottleneck links on a path. We assume this to be 10.

- The proposed changes are acceptable to B if no s_l is negative. The protocol discussed in Appendix A.3.1 is used to determine this. To prevent A from knowing the signs of each s_l, B calculates $s_l r_l$ where r_l is ± 1 with equal probability. A few dummy values may also be used by B to prevent A from knowing the exact number of bottleneck links. The computation complexity is $O(N)$ decryptions and communication complexity is $O(N)$.

- A sends $E_A(1)$ if $s_l r_l$ is positive and $E_A(-1)$ otherwise. The communication complexity is $O(N)$. Since A can precompute $E_A(\pm 1)$, there is no encryption complexity. Zero-knowledge proofs such as those in [9] may be used here by B to verify that the received numbers encrypt ± 1.

- B multiplies $E_A(sign(s_l r_l))$ with r_l to generate an encrypted vector containing a -1 iff the corresponding bottleneck link would get congested and 1 otherwise. The primitive described in Section A.3.2 can be used by A to determine if there is a -1, i.e., if there is a link that might get saturated. $O(N)$ multiplications are required here.

Typical values of D, P are about 200 and 10 [30]. We assume N to be 50 . Assuming that specialized hardware can perform encryption/decryption operations in a few hundred microseconds and multiplication in a few microseconds, the above protocol can be easily executed in less than a minute. This is reasonable considering that traffic engineering changes need not be done more than once an hour[40]. If the participants follow the protocol, the above protocol is secure because of two reasons. The first is that B cannot determine the value of any encrypted value. Thus, it gets to know only the outcome, from A. The second reason is that A knows nothing about the signs of s_ls because of the r_ls. Hence, A knows only the outcome.

Wrong inputs may be provided to cause two kinds of wrong outcomes. They could cause unacceptable changes to be accepted in which case the resulting congestion will be observed by the other AS. Out-of-band mechanisms could be used to "replay" the protocol and determine the malicious AS. But, if an acceptable change is rejected, this protocol has to be retried. A malicious A or B could obtain information about the other AS using multiple tries. This is a fundamental limitation of this protocol. Next, we develop a protocol that does not suffer from this limitation.

4.2 Optimal Exit Points

We now consider a generalization of the goals of the above protocol using a linear programming formulation. Assuming that the paths from ingress to egress in A and egress to ingress in B are fixed, how can A and B cooperatively determine the best exit point (from A to B) for flows. This determination must be done without either AS revealing its confidential bandwidth constraints. Other issues such as bidirectional traffic, minimization of the traffic asymmetry at peering points, minimization of path inflation and other goals may also have to be considered, in reality.

Using the notations introduced in Table 1, our goal is to determine $x_{hj} \geq 0$, the amount of flow f_h exiting at peering point p_j. The linear programming problem for the corresponding max-flow problem can be written using the flow conservation and bandwidth constraints of A and B as:

$$maximize \sum_{h,j} x_{hj} \;\; given\ that \tag{3}$$

$$\sum_{j=1}^{P} x_{hj} \leq c_h \; \forall\, h. \tag{4}$$

$$\sum_{h=1}^{F} \sum_{j=1}^{P} \alpha_{hjk} x_{hj} \leq a_k \; \forall\, k. \tag{5}$$

$$\sum_{h=1}^{F} \sum_{j=1}^{P} \beta_{hjl} x_{hj} \leq b_l \; \forall\, l. \tag{6}$$

Using $X^t = (x_{11}, \ldots, x_{1P}, \ldots, x_{FP})$, this is equivalent to maximizing $c^T X$, where c is a column vector of size FP all of whose entries are 1, given $(VX \leq W)$ (see Figure 4.2).

A method to solve linear programming problems securely was presented in [10]. This cannot be used directly by us since V and G are not arbitrary matrices over a field (both are integral, for instance). We use a modified version of their technique in our protocol and also provide a proof why this protocol is secure.

The basic observation in [10] that we leverage is that X, the solution of the above linear program also satisfies $X = QX'$ where X' (a non-negative vector) maximizes $c'^T X' = (c^T Q) X'$ given constraints $V'X' = (PVQ)(Q^{-1}X) \leq PW = W'$. Here P, Q^{-1} are arbitrary invertible matrices consisting of positive entries (because, multiplying a constraint with -1 reverses the inequality). To eliminate the requirement of positive entries, we augment each constraint with a dummy variable (which must also be non-negative) and make the inequality into an equality. For convenience, we abuse notation and refer to the system of equalities as $VX = W$. Under the same objective function, the solution to this equality is the same as the original set of inequalities. In this system of equalities, V has $r = F + M + N$ rows and $c = FP + F + M + N$ columns.

Figure 2: The LP problem $VX \leq W$ for determining optimal exits of flows from A to B.

$$
\begin{array}{c}
\begin{array}{ccccc} x_{11} & \dots & x_{1P} & \dots & x_{FP} \end{array} \\
\left(\begin{array}{ccccc}
1 & \dots & 1 & \dots & 0 \\
0 & \dots & 0 & \dots & 0 \\
\dots & \dots & \dots & \dots & \dots \\
0 & \dots & 0 & \dots & 1 \\
\alpha_{111} & \dots & \alpha_{1P1} & \dots & \alpha_{FP1} \\
\dots & \dots & \dots & \dots & \dots \\
\alpha_{11M} & \dots & \alpha_{1PM} & \dots & \alpha_{FPM} \\
\beta_{111} & \dots & \beta_{1P1} & \dots & \beta_{FP1} \\
\dots & \dots & \dots & \dots & \dots \\
\beta_{11N} & \dots & \beta_{1PN} & \dots & \beta_{FPN}
\end{array} \right)
\end{array}
X \leq
\left(\begin{array}{c}
c_1 \\ \dots \\ c_F \\ a_1 \\ \dots \\ a_M \\ b_1 \\ \dots \\ b_N
\end{array} \right)
$$

Our protocol uses the above observation to have AS B calculate a new system of equalities with $V' = PVQ$, $W' = PW$, $c'^T = c^T Q$ where P, Q are arbitrary invertible rational matrices chosen by B. B calculates this new system using homomorphic operations on (encrypted) entries sent by A. The new system is decrypted by A which calculates X', the solution to the new system. $X = QX'$ is then calculated by B. Only the outcome, the plaintext version of X, is revealed to B. Hence, if the cryptosystem is secure, then B cannot deduce anything other than from the outcome. However, A gets the plaintext versions of V', W', c', X' and X. The following theorem states the conditions under which these do not provide any information to A.

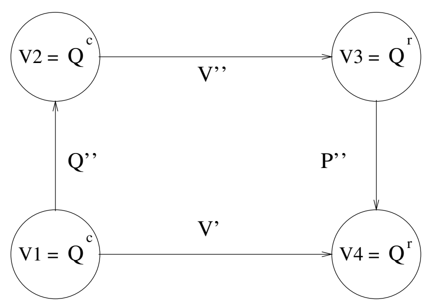


Figure 3: Vector Space Transformation of $VX = W$. Q^d denotes a vector space over rationals of dimension d. V', V'' are rational matrices of rank r with r rows and c columns ($r > c$). P'' and Q'' are invertible matrices with r and c rows/columns respectively. Given V', simple diagram chasing using the bases of the vector spaces ensures the existence of P'' and Q'' for all V''.

Theorem: Given V', W', c', X', X, A can obtain no information about V, W if there exist rational P'', Q'' for all V'', W'', c'' such that $P''V''Q'' = V'$, $P''W'' = W'$, $c' = Q''^T c''$ and $X = Q''X'$. P'', Q'' exist if (1) V' and V'' are both rational matrices of rank r (2) c'' and X' are linearly independent, and (3) W'' is linearly dependent on $V''X, V''c'$.

Proof: A rational matrix (with x rows and y columns) of rank $t \leq min(x, y)$ represents a vector space transformation from a vector space of dimension y to a vector space of dimension x where the null space is of dimension $y - t$. Invertible matrices with x rows and columns have rank x. Both left and right multiplication of matrices compose two vector space transformations to generate a new vector space transformation. In our case, the composition of transformations $P''V''Q'' = W'$ is shown in Figure 3. Here, Q^d represents a vector space of dimension d over the field of rationals.

Given V' and V'' in Figure 3 it is easy to construct P'' and Q'' such that $P''V''Q'' = V'$: Choose the same basis for V_1 and V_2 and take arbitrary Q'' such that the null space of V' maps to the null space of V''. V' and V'' map the rest of the bases to bases in V_3 and V_4 respectively. Now, ensuring that the basis element $\vec{v}_1$ of V_1 maps onto the $V'(\vec{v}_1)$ under $P''V''Q''$ implies that P'' maps $V''Q''(\vec{v}_1)$ to $V'(\vec{v}_1)$. Note that, such consistency requires that V' and V'' have the same rank. A knows W', c', X' and X too. Any candidate set V'', W'', c'' (and X'') impose additional constraints on P'' and Q''. Since $c' = Q''c''$ and $X'' = Q''X'$ represent requirements of Q'' which was arbitrary in the above construction, they can be accommodated unless c'' and X' are not linearly independent. Also, $W' = P''W''$ specifies the action of P'' on one vector which would not contradict Q'' unless W'' is linearly dependent on $V''X, V''c'$. □

The above proof motivates the design of our protocol. B first converts the system of inequalities into a system of equalities. Small random noise is added to V, W, c so that V is of rank r with high probability. The noise also ensures that there is negligible probability of either of the two dependence conditions mentioned above. Hence, the conditions of the above theorem are satisfied with high probability. B then chooses arbitrary invertible P and Q to generate a new linear programming problem which A solves. B uses a homomorphic encryption scheme to generate the new LP problem and hence, knows nothing about As inputs. The new system does not reveal any information on Bs inputs to A since it could be generated from any original system that satisfies the conditions of the above Theorem. Note that the addition of random noise to the constraints should not have significant effect on X. There is evidence for this; Roughan et al. [34] showed that the estimated traffic matrices (which contain small estimation errors, akin to random noise) produce good enough results in (intra-domain) traffic engineering.

Our protocol is based on the above observations and homomorphic encryptions to construct V', W', c'. Note that the method to construct V', W', c' proposed in [10] cannot be used here since V and W satisfy special properties (they have only boolean entries, for instance).

Protocol Description:

- A transfers to B, the rows of V and W that are known only to it by encrypting them with its public key.
- B creates the matrix V and W with A's encrypted entries and its unencrypted constraints. The entries of these matrices are scaled randomly (rows or columns at a time, to preserve the equality constraints) and random noise (small integers) added to convert V into a matrix of rank r.
- For the proof of security to be valid, random invertible rational P and Q are chosen. V', W', c' can all be calculated using homomorphic encryption since the only arithmetic operations required are multiplication by known factors and addition of ciphertexts which can be done with homomorphic cryptosystems. Observe that, since only integers (not rationals) can be encrypted, operations with rationals might be performed by remembering the plaintext denominator or by randomly scaling P, Q so that they are integral.
- V', W', c' are transferred to A which solves the system of equalities and sends X' to B. B uses $X = Q^{-1}X'$ to A.

Compared to solving the original set of inequalities, the above protocol has extra complexity due to two factors. First, by converting the inequalities into equalities, we increase the size of the LP problem. Second, the processing of encrypted data by B to generate V', W', c' also increase the complexity. The former can be alleviated by using fast LP problem solvers. For the latter, all inputs of A are used to calculate every entry of the V'. Thus $O(FPM)$ entries need to be multiplied $O((FP + M + N)(M + N))$ times. The numbers specified earlier indicate this to take millions of operations, at least for large tier-1 ISPs. Hence, in such cases, flows may have to be restricted to certain peering points thereby reducing P, M and F. Subsets of flows would then have to be solved independently. Investigation of the impact of such techniques on optimality is a subject of future work to us. We end this section by noting that the above method can be applied to solve other LP problems provided that these are not sensitive to the random noise that needs to be added to achieve confidentiality.

5 Sharing Policy Information in BGP

In this section, we illustrate the need for sharing routing policies between ASs using the well-known problem of divergent routing caused by conflicting policies of ASs [41, 24] and propose a protocol that determine if divergence is present in the context of confidential policies.

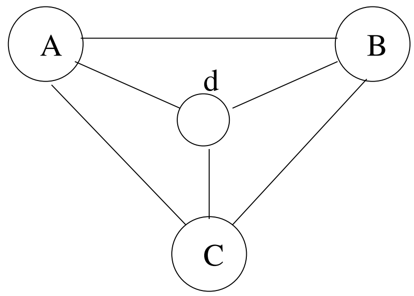


Figure 4: Divergence-causing AS topology when A prefers B, B prefers C, C prefers A to reach destination d.

Oscillations that occur due to unintended interaction of BGP policies was first shown in [41]. Figure 4 shows a simple AS topology that can cause divergent routing. This is because, each A prefers B, B prefers C and C prefers A to reach destination d. A more thorough analysis of policy-induced divergence properties was done in [24]. One approach proposed to remedy this situation [21] is to have routing registries where all the routing policies are revealed and static analysis is done to verify safety. The set of guidelines proposed in [19] gives a way to constrain local policies so as to make BGP policies safe although the use of backup routing does require some global coordination. Another way of guaranteeing safety is by enabling run-time detection of divergence using route histories, extensions to BGP, as proposed in [22]. However, route histories leak information on local policies [13]. A proposal to anonymize route histories is provided in [22] but its efficacy is unclear. Currently, it is hoped that such divergence will not happen because ASs use the constrained local policies described in [19].

Static analysis of AS policies to check for divergence is impractical because many ASs are unwilling to reveal their internal policies. Another reason why it is impractical is that this problem is NP-complete [24]. We take such a scheme proposed in [23] and show how it can be implemented without requiring ASs to reveal their policies. While we do not solve the computational inefficiency of the solution, our aim is to demonstrate that the only barrier to static analysis is the development of policy specifications (for BGP and other policy-aware MPDSs that could experience policy-induced anomalies) that can be efficiently analyzed without being as constraining as those proposed in [19]. Additionally, our protocol could be used to analyze the presence of a dispute wheel for continuously flapping prefixes, i.e., prefixes that exhibit dispute wheel-like behavior.

The scheme for static analysis that we use was proposed by Griffin et al. in [23]. They show that the absence of *dispute wheels* is a sufficient condition for BGP convergence. They define a dispute wheel for a destination d as a set $\Pi = (U, Q, R)$ of size k, where $U = (u_0, ...u_{k-1})$ is a sequence of nodes, $Q = (Q_0, ...Q_{k-1})$ and $R = (R_0, \ldots R_{k-1})$ are sequences of paths such that $\forall i, 0 \leq i \leq k-1$, (1) R_i is a path from u_i to u_{i+1}. (2) Q_i a path permitted by the policy of node u_i to destination d. (3) R_iQ_{i+1} is also a path permitted by the policy of node u_i. (4) Node u_i prefers the path R_iQ_{i+1} over Q_i. We now describe a protocol that can use confidential policies to check if a given candidate dispute wheel, Π is a dispute wheel or not.

Protocol Description: Each AS u_i in this candidate dispute wheel sets its input a_i to be 0 if all 4 conditions above are true and to 1 otherwise. The candidate dispute wheel is a dispute wheel if all a_is are 0. This can be achieved using a threshold cryptosystem in which the key shares can be computed using an out-of-band mechanism. A random u_i initiates the protocol by sending $E(a_i)$ to u_{i+1}. Any subsequent AS u_j blinds the received value if a_j is 0 and replaces it with $E(1)$ if not. Thus, the final value $E(A_\Pi)$ encrypts 0 iff Π is a dispute wheel. This value is decrypted collectively. A decrypted value of 0 indicates that Π is a dispute wheel. The number of outcomes can be reduced by determining the existence of a dispute wheel among $n > 1$ candidates Π_j. This can be done by multiplying $E(A_{\Pi_j})$ and decrypting the result to get $\sum_j A_{\Pi_j}$. If this is not n, then there exists a dispute wheel. Note that, this process essentially computes the "OR" of the "AND" of boolean variables. On determining the presence of a dispute wheel, the ASs involved would modify only the relevant policies to make them locally constrained as specified in [19].

Any one participant could misbehave to falsely indicate the presence or absence of a dispute wheel. This could simply be done by providing wrong inputs (especially to indicate the lack of conflict). As mentioned in Section 2.2, we believe that such misbehavior is not realistic since no provider would willfully cause divergence of a shared infrastructure.

6 Potential Applications

In this Section, we illustrate how sharing both operational conditions and policy information can solve problems that arise in inter-domain routing and MPDSs such as CDNs and policy-based resource allocation. These represent broad areas of future work for us. Where applicable, we discuss our preliminary solutions.

6.1 Peering among Content Distribution Networks (CDNs)

In [5], the authors propose mechanisms to allow a CDN A to offload client workload to peer CDNs, such that the Service Level Agreement (SLA) of each client is satisfied and the costs of doing so are minimized. In this case, since the redirection policies of CDNs are confidential, whether a peer CDN can satisfy the SLA (maximum server-client delay) is not known. In [5], the solution proposed is to deduce this information using measurements. A disadvantage of this approach is that there is a cost associated to wrongly determining if a peer CDN can satisfy a set of clients. In Appendix B, we describe an SMPC technique by converting the problem to be one of linear programming.

6.2 Inter-domain Path Selection

Choosing a path of good quality from a source to destination is a problem that arises repeatedly in networks. Primitives involving path quality may be used by the infrastructure (e.g., to construct QoS routing tables) or by users (e.g., multi-homed endhosts who need to choose the best outgoing interface for traffic). End-to-end measurement techniques may suffice for the latter, at least in the near future since there is not much demand for per-flow path setup. However, there is a great need for better path selection mechanisms at the AS level [6, 27].

Since most paths in the Internet traverse multiple domains and the quality of intra-domain links is considered confidential, SMPC-based mechanisms could be extremely useful for path selection. Comparing two paths or determining if a path can support a desired level of quality are two useful primitives for path selection. Path quality itself might be measured using additive metrics such as (logarithm of) loss, delay, jitter or minimum/maximum metrics such as available bandwidth. Determining if the sum/minimum/maximum of the inputs from different entities is larger than another value, without revealing the inputs, would be of great use in considering path quality in inter-domain routing. For instance, each step of Bellman-Ford algorithm for shortest path computation essentially compares the "length" of two paths in the graph. Since typical AS paths are $3 - 4$ hops long, the outcome of such comparisons (over the whole execution of Bellman-Ford) might leak more information than desirable. Eliminating such leaks is a subject of future work for us.

6.3 Policy-driven MPDSs for Computational Resources

Recent research [16, 18, 25] has focused on large-scale distributed infrastructures offering resources such as computing, storage etc. The notion of utility computing, large-scale sharing of computing resources, has received a lot of attention from industry. Resource allocation in these systems, which are likely to have multiple providers (the various universities of Planetlab [31] could be considered as providers), is very important and must consider the policies (i.e., preferences) of these providers. To our knowledge, how confidentiality of resource allocation policies may be enforced while ensuring that resource allocation be efficient and near-optimal, has not been dealt with in prior research.

7 Conclusions and Future Work

Many large scale distributed systems involve multiple providers. Managing, optimizing and troubleshooting these systems requires the use of confidential information from different providers. In this paper, we provide SMPC-based techniques that allow such cooperation in inter-domain routing without requiring that such information be made public. We propose two techniques to illustrate the sharing of a network's operational conditions for better traffic engineering. We also show that determining policy-induced routing divergence does not require that policies be revealed. In designing our techniques, we leverage key characteristics of MPDSs. The out-of-band relationship between providers removes the need to consider unfairness due to abnormal protocol termination. Since all providers have reason to keep the system healthy makes many adversarial models unrealistic. However, adversarial behavior to determine private information of other providers is certainly possible and must be guarded against. We also discuss potential applications of SMPC-based techniques and preliminary solutions for a host of other problems including path selection in routing, client redirection in CDNs and policy-based resource allocation. These are areas of future work to us. MPDSs pose challenges that can be solved using SMPC-based techniques. We hope that our work encourages future MPDS designs to use such mechanisms instead of making them opaque and hence, prone to inefficiencies and failures.

References

[1] Secure Multiparty Computations. `http://www.tcs.hut.fi/~helger/crypto/link/mpc/`.

[2] S. Agarwal, C.-N. Chuah, and R. H. Katz. OPCA: Robust Interdomain Policy Routing and Traffic Control. In *Proc. of IEEE OpenArch*, 2003.

[3] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In *Proc. of ACM SIGMOD*, 2003.

[4] D. "Allen B. Using Pathchar to Estimate Internet Link Characteristics. In *Proc. of ACM SIGCOMM*, 1999.

[5] L. Amini, A. Shaikh, and H. Schulzrinne. Effective Peering for Multi-provider Content Delivery Services. In *Proc. of IEEE INFOCOM*, 2004.

[6] O. Bonaventure, B. Quotin, and S. Uhlig. Beyond Inter-domain Reachability. In *Proc. of Workshop on Internet Routing Evolution and Design (WIRED)*, 2003.

[7] C. Cachin. Efficient Private Bidding and Auctions with an Oblivious Third Party. In *Proc. of ACM Conference on Computer and Communications Security (CCS)*, 1999.

[8] C. Cachin and J. Camenisch. Optimistic Fair Secure Computation (Extended Abstract). In *Proc. of Advances in Cryptology (CRYPTO)*, 2000.

[9] I. Damgard, M. Jurik, and J. B. Nielsen. A Generalization of Paillier's Public-Key System with Applications to Electronic Voting, 2003. Unpublished Report.

[10] W. Du. *A Study of Several Specific Secure Two-party Computation Problems*. PhD thesis, Purdue University, West Lafayette, Indiana, 2001.

[11] T. El Gamal. A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms. In *Proc. of Advances in Cryptology (CRYPTO)*, 1984.

[12] D. Estrin, Y. Rekhter, and S. Hotz. RFC 1322 - A Unified Approach to Inter-domain Routing, May 1992.

[13] N. Feamster and H. Balakrishnan. Towards a Logic for Wide-Area Internet Routing. In *Proc. of ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA)*, August 2003.

[14] N. Feamster, J. Borkenhagen, and J. Rexford. Guidelines for Interdomain Traffic Engineering. *ACM SIGCOMM Computer Communications Review (CCR)*, October 2003.

[15] J. Feigenbaum, N. Nisan, V. Ramachandran, R. Sami, and S. Shenker. Agents' Privacy in Distributed Algorithmic Mechanisms, 2002.

[16] I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *Lecture Notes in Computer Science*, 2001.

[17] P.-A. Fouque, G. Poupard, and J. Stern. Sharing decryption in the context of voting or lotteries. *Lecture Notes in Computer Science*, 1962, 2001.

[18] Y. Fu et al. SHARP: An Architecture for Secure Resource Peering. In *Proc. of ACM Symposium on Operating Systems Principles(SOSP)*, 2003.

[19] L. Gao and J. Rexford. Stable Internet Routing without Global Coordination. In *Proc. of ACM SIGMETRICS*, June 2000.

[20] G. Goodell et al. Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing. In *Proc. of Network and Distributed System Security (NDSS) Symposium*, San Diego, CA, February 2003.

[21] R. Govindan, C. Alaettinoglu, G. Eddy, D. Kessens, S. Kumar, and W. Lee. An Architecture for Stable, Analyzable Internet Routing. *IEEE Network Magazine*, January/February 1999.

[22] T. Griffin and G. T. Wilfong. A Safe Path Vector Protocol. In *Proc. of IEEE INFOCOM*, March 2000.

[23] T. G. Griffin, F. B. Sheperd, and G. Wilfong. The Stable Paths Problem and Interdomain Routing. *IEEE/ACM Transactions on Networking*, 10(2), April 2002.
[24] T. G. Griffin and G. Wilfong. An Analysis of BGP Convergence Properties. In *Proc. of ACM SIGCOMM*, 1999.
[25] J. Kubiatowicz et al. OceanStore: An Architecture for Global-Scale Persistent Storage. In *Proc. of Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, November 2000.
[26] K. Lai and M. Baker. Measuring Link Bandwidths Using a Deterministic Model of Packet Delay. In *Proc. of ACM SIGCOMM*, 2000.
[27] R. Mahajan. Negotiation-based Routing. In *Proc. of Workshop on Internet Routing Evolution and Design (WIRED)*, 2003.
[28] M. Mitzenmacher. Compressed Bloom Filters. In *Proc. of ACM Symposium on Principles of Distributed Computing (PODC)*, 2001.
[29] P. Paillier. Public-Key Cryptosystems Based on Discrete Logarithms Residues. In *Proc. of Eurocrypt*, 1999.
[30] K. Papagiannaki, N. Taft, and C. Diot. Impact of Flow Dynamics on Traffic Engineering Design Principles. In *Proc. of IEEE INFOCOM*, March 2004.
[31] PlanetLab, 2004. `http://www.planet-lab.org`.
[32] Professional Projects Company, 2003. `http://www.proproco.co.uk/million.html`.
[33] Y. Rekhter and T. Li. RFC 1771 - A Border Gateway Protocol 4(BGP-4), March 1995.
[34] M. Roughan, M. Thorup, and Y. Zhang. Traffic Engineering with Estimated Traffic Matrices. In *Proc. of Internet Measurement Conference(IMC)*, 2003.
[35] S. Singh et al. The Case for Service Provider Deployment of Super-Peers In Peer-To-Peer Networks. In *Proc. of International Workshop of Economics of Peer-To-Peer Systems*, 2003.
[36] N. Spring, R. Mahajan, and T. Anderson. Quantifying the Causes of Path Inflation. In *Proc. of ACM SIGCOMM*, August 2003.
[37] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP Topologies with Rocketfuel. In *Proc. of ACM SIGCOMM*, 2002.
[38] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the Internet Hierarchy from Multiple Vantage Points. In *Proc. of IEEE INFOCOM*, 2002.
[39] E. Teske. Square-root Algorithms for the Discrete Logarithm Problem, 2001.
[40] S. Uhlig, V. Magnin, O. Bonaventure, C. Rapier, and L. Deri. Implications of the Topological Properties of Internet Traffic on Traffic Engineering. In *Proc. of ACM Symposium on Applied Computing(SAC), Special Track on Computer Networks*, March 2004.
[41] K. Varadhan, R. Govindan, and D. Estrin. Persistent Route Oscillations in Inter-Domain Routing. *Computer Networks*, March 2000.
[42] F. Wang and L. Gao. On Inferring and Characterizing Internet Routing Policies. In *Proc. of Internet Measurement Conference (IMC)*, October 2003.
[43] J. Winick, S. Jamin, and J. Rexford. Traffic Engineering between Neighboring Domains, July 2002. `http://www.research.att.com/~jrex/papers/interAS.pdf`.
[44] A. C. Yao. Protocols for Secure Computations. In *Proc. of Foundations of Computer Science (FOCS)*, 1982.

APPENDIX

A Cryptography

A.1 Details of Homomorphic Encryption Schemes

Table 2 provides a brief overview of El Gamal and Paillier's cryptosystems. As can be seen, the former is multiplicative homomorphic while the latter is additive homomorphic.

For small x, determining discrete logarithm is not hard, i.e., given g, g^x x can be determined using the "baby-step giant-step" algorithm which trades off memory for time. A survey of such methods is provided in [39]. Hence by modifying El Gamal to encrypt g^m instead of m, and requiring the additional discrete logarithm step for decryption, El Gamal can be additive homomorphic, under the assumption of small plaintexts only.

A.2 Commutative Encryption

Our primitive uses commutative encryption [3]. Two encryption functions f_1, f_2 are commutative if $f_1(f_2(x)) = f_2(f_1(x))$. For suitably chosen prime p, any two discrete-log based encryptions [11] defined as $f_y(x) = x^y(mod)n$ satisfy this property because $(x^{y_1})^{y_2}mod(p) = (x^{y_2})^{y_1}(mod)p$.

Table 2: Encryption Techniques

Name	El Gamal	Paillier
Group Used	Z_p where p is a large prime	Z_{n^2}, n as in RSA
Public Key	$(p, g, y = g^x)$ g a generator $1 < x < p$ is random	(n, g) s.t. $order(g) = n\alpha$ $1 < \alpha < \lambda = LCM(p-1, q-1)$
Private Key	x	(p, q)
Ciphertext c of plaintext m	$c = (a, b) = (my^k, g^k)$, k is random	$c = g^m r^n$ $0 < r < n$ is random
Plaintext m of ciphertext c	$m = ab^{-x}$	$m = \frac{L(c^\lambda)}{L(g^\lambda)}$ where $L(u) = (u-1)/n$
Security Assumption	Hardness of discrete logarithm	Hardness of computing the order
Encryption/ Decryption Complexity	$O(log(k))$/ $O(log(x))$	$O(log(n))$/ $O(log(\lambda))$

A.3 Useful Primitives for Multi-Party Computations

In this subsection, we construct simple cryptographic primitives that we use in the paper. We would like to emphasize that our constructions, while reasonably efficient are not the best that can be done. More efficient constructions should be possible.

A.3.1 Comparing Two Scalars

The first primitive that we develop is for A to know the bigger of two scalars a and b private to two parties A and B respectively under the assumption that $x = (a - b)$ such that $|x| \leq x_{max}$. This problem is a special case of the Yao's millionaires' problem [44] without fairness requirements (unlike [7]) and our solution is more efficient than the solution in [32].

The basic idea behind our solution is for A to verify the membership of $x = (a - b)$ in the set of integers from $[0, x_{max}]$ securely. This is done as follows:

1. A sends $E_A(a)$ to B, where E_A denotes encryption with As public key.
2. B calculates $E_A(a - b + r)$ for some random r using the homomorphic property and sends it to A.
3. A and B each choose encryption functions f_A and f_B respectively that commute with each other (see Section A.2. B calcuates the set $Pos_B = (f_B(r), f_B(r+1), \ldots, f_B(r + x_{max}))$ and sends a randomly permuted Pos_B to A.
4. A decrypts $x' = (a - b + r)$ and sends $f_A(x')$ to B.
5. B calculates $f_A \circ f_B(x')$ and sends it to A.
6. A checks if it belongs to the set $Pos = (f_A \circ f_B(r), \ldots, f_A \circ f_B(r + x_{max}))$.

If the above protocol is followed, then B cannot determine a since E_A is secure. A knows x' but the random r prevents it from knowing anything about x or b. B knows $f_A(x')$; it is computationally hard for B to calculate x' (and hence, a) from $f_A(x')$. Similarly, the hardness of discrete logarithm prevents A from knowing the value of r from Pos_B. To prevent either party from using previous comparison operations (for instance, the same $f_A(x')$ in two operations would imply that the difference in the a values of the two operations is the difference between the b values), A and B should choose a different set of commutative encryption functions for each comparison.

The above protocol can be made more efficient since A knows the inverse of f_A, $f_A^{-1}(z) = z^{(y_A^{-1})}$ which satisfies $f_A^{-1} \circ f_B \circ f_A(z) = f_B(z)$. This allows A to calculate $f_B(x')$ which can directly be compared with Pos_B. Also, B can pre-compute Pos_B and use efficient compression tools such as compressed Bloom filters [28] to send the membership vector to A. Thus, the whole comparison requires one encryption and decryption, three exponentiations.

A.3.2 False Component of a Boolean Vector

This primitive is for two parties, A and B, to determine if a boolean (i.e., consists of 1's and 0's only) set $V = (v_1, \ldots v_n)$ contains at least one 0. V is assumed to be encrypted with A's public key and calculated by B using As private inputs. This can easily be done by having B calculate $E_A(\sum_i v_i - n) = E_A(x)$. The vector contains a 0 iff x is negative. The sign of x can be determined using the comparison primitive discussed in Section A.3.1. This primitive can also be used to determine if a vector with only 1's and -1's has any -1's. Note that, a malicious B can provide any value it chooses for decryption to A. Thus, it can determine the sign of exactly one combination of As private inputs.

B CDNs

Now, we describe a technique that allows a CDN to calculate optimal redirection criteria using the private inputs of peer CDNs. We assume that there are C client sets c_i and S server sets s_j. m_i denotes the maximum workload of client set c_i and u_j denotes the cost per unit workload of server set s_j. The quantities to be determined are x_{ij}, the workload of c_i that will be offloaded to s_j. The confidential inputs of a server set are the remaining capacity of a server, C_j and D_{ij}, which is 0 if server s_j cannot satisfy the SLA for client c_i and 1 otherwise. The assignment of clients to servers is obtained by calculating x_{ij} under the constraints of the following linear programming problem:

$$minimize \sum_j u_j \sum_i x_{ij} \tag{7}$$

$$for\ the\ maximum \sum_{i,j} x_{ij}\ \ such\ that \tag{8}$$

$$\sum_j x_{ij} \leq m_i \tag{9}$$

$$\sum_i x_{ij} \leq C_j \tag{10}$$

$$\sum_i (1 - D_{ij}) x_{ij} \leq 0 \tag{11}$$

The objective function is to minimize the cost, when the maximum number of clients are serviced. The constraints are concerning the maximum workload per client, capacity of a server set and the inability of the server set to service some client sets.

We can adapt the technique in Section 4.2 for use with multiple participants. The basic idea is that each CDN S_j involved would choose an arbitrary P_j and Q_j. Denoting the above LP problem to be $VX = W$, the CDNs would collectively calcute $V' = (\sum_j P_j)V(\sum_j Q_j)$, $W' = (\sum_j P_j)W$ (the objective functions are similarly transformed). To calculate V', each CDN must first advertise P_jV, calculate $(\sum_j P_j)V$ and multiply it with Q_j and finally advertise this matrix. These advertised matrices, when added up, give an encrypted version of V'. Threshold variants of homomorphic schemes such as Paillier's [9, 17] are used to work with encrypted matrices.

The amount of encrypted arithmetic to be performed by each CDN is comparable to the scheme in Section 4.2. However, the use of threshold cryptography etc. increase the communication complexity of this approach. It is an open issue if the complexity can be reduced to manageable levels. A misbehaving CDN could cause the wrong solution to be reached to obtin more clients. A wrong solution that does not satisfy constraints of honest CDN(s) is detectable, though. Characterizing the probability of detecting misbehavior and developing techniques (maybe, zero-knowledge proofs) to catch the misbehaving CDN are the subject of future work.