Securing the Internet Metering and Billing

Marcelo Pias [1] Steve Wilbur [1] Saleem Bhatti [1] Jon Crowcroft [2]
[1] University College London (UCL), Computer Science Dept.
Gower Street, WC1E 6BT, London, UK
[2] University of Cambridge, Computer Laboratory
JJ Thomson Avenue, CB3 0FD, Cambridge, UK

Abstract—**In the near future, billing for network services will not only be concerned with time or volume based accounting but also in ways of measuring the quality of the service provided. Dynamic price schemes such as congestion-based charging have been proposed. In some of these models, the charging infrastructure relies on the distribution of electronic tariffs to end-users machines. The tariff structure includes the price information and an algorithm to calculate the charge. Thus, the monitoring of the network usage according to this tariff is essential within these frameworks.**

However, little attention has been given to the security issues associated with Internet metering in these schemes. This has had a great impact on the new models proposed today, since security has become a major concern in open networks. Systems that naturally have incentive to frauds such as metering systems used for billing purposes must deal with security threats in large scale environments.

This article compiles the security issues of a dynamic networked system where electronic tariffs and Service Level Agreements (SLA) structures are distributed among service providers and customers. To address these issues, a set of security protocols are outlined.

Keywords: security, QoS network metering, billing for communication services

I. Introduction

Several approaches and architectures have been proposed which deal with the means to offer QoS in the Internet. Such a spectrum ranges from prioritising classes of aggregate traffic (DiffServ) [1] to mechanisms that reserve resources to match the QoS requested (IntServ) [2]. Within the same spectrum, new approaches are being put forward which propose controlling the Internet application behaviour using different price schemes [3] [4] [5].

In this case, billing is not only concerned with the traditional time or volume based accounting but also in QoS measurements. This trend gives further motivation to incorporate metrics for quality of service into metering systems. In [6] a differential charging framework is proposed that relies on electronic tariffs. Such a tariff structure includes the price information and an algorithm to calculate the charge.

This article assumes the distribution of electronic tariffs and service level agreements (SLA) structures among service providers and customers. This dynamic environment requires the configuration of specialised components, called network meters, which are responsible for taking the measurements required by the tariffs/SLAs [7]. For instance, a telecommunication service (e.g. IP Telephony) should have a service specification composed by a SLA and tariff. For this case, the voice session is measured to guarantee the service level specified in the SLA structure and to charge it according to the tariff in use.

The business model, considered in this article, poses a set of metering requirements (Fig. 1). Such a canonical model considers business roles and their relationships in a multiservice network environment. The roles consist of (a) **network providers** (bulk network provider, network wholesaler and network retailer) who sell a network-based service (e.g. data transportation, network access); (b) an **application-level service provider** who sells services such as audio/video streaming and IP Telephony; (c) a **network management service provider** which includes meter manufacturers who sell services such as QoS metering for billing; and (d) **customers** who subscribe to a service requiring a certain level of QoS. Most significantly for this article is how such these business roles interact with each other. Such interactions raise security issues that need to be addressed.

This project is funded by British Telecom Research Labs (BTExact), Ipswich, UK

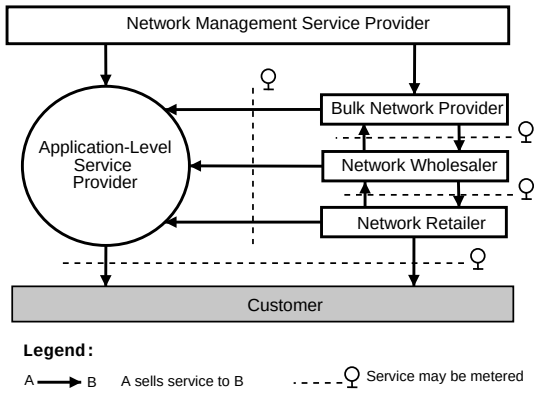

Fig. 1. Business Model

As with utility metering, there are incentives for customers or even providers to cheat. Such fraud may lead to drastic results when achieved in large scale. Generally, this paper has two goals. The first one is to prevent frauds committed by malicious customers (e.g. meter tampering). The second goal is to protect customers of any damage that an installed meter code may cause to its system.

A lesson to be learnt from utility metering is pointed out by *Ross Anderson* in [8]. It says that the real security breaches in electricity prepayment metering resulted from bugs. Most interesting, they were discovered by accident and exploited in large scales, costing fortunes to fix. Although utility metering is a simpler scenario than the one envisaged by the current work, it exemplifies the issue concerning metering for billing people.

This article analyses what can go wrong with network metering for advanced services such as IP Telephony with focus on the security aspects. To accomplish the level of security required, a model to secure the Internet QoS metering is proposed. Section II clarifies the problem addressed in this work. Section III outlines a list of security requirements that have to be fulfilled. A set of protocols is presented within section IV in order to match such requirements. Finally, the conclusions and suggestions for future work are presented in section V.

II. Problem Description

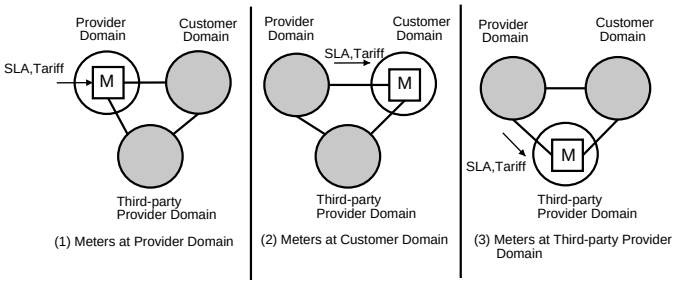

Fig. 2. Basic Metering Scenario

The problem investigated mainly arises from the requirements posed by the business model shown in Fig. 1. Such a model suggests that metering should be performed between the providers and customers

interfaces. Fig. 2 introduces the generic scenario considered.

First of all, measurement points are very flexible. When looking from the left side of Fig. 2, QoS meters (marked as 'M') may be installed at the provider's domain (1). Such arrangements seem to be the standard practise at the present.

Alternatively, customers may have meters running in their own domain (2). Such a case, not very often deployed today, puts the customer in the metering loop. On the one hand, this seems unacceptable to some providers. On the other hand, it is likely to become the future scenario where the data is collected and summarised at the end-user. This is also in line with the nature of the Internet dictated by the end-to-end argument [9]. In synthesis, it argues that the intelligence should be placed at the edges of the network leaving the core as simple as possible. Having said that, it seems natural to deploy meters to the end users.

Additionally, the metering architecture could be deployed in the network itself using third-party provider domains (3). A mobile IP scenario is one of the examples of this third-party metering position. It might be the case in such an example that a home provider makes an agreement with a foreign provider to measure its customer traffic according to the original service level agreement (SLA).

Finally, other meter position arrangements are also possible. Thus, meters at the provider domain (1) can measure traffic simultaneously with meters at the customer domain (2). This latter scenario is likely to happen when providers and customers do not trust each other.

While the flexibility of varying the metering points accounts for the key feature in the scenario presented in Fig. 2, it does introduce problems. Such issues concern questions of (a) how to configure the metering system, once it is installed, with the SLA and tariff structures; (b) what threats a metering system may be exposed to; (c) finding the limitations of Internet QoS metering in terms of performance; and (d) finding suitable positions for metering. Therefore, a hybrid metering architecture called EdgeMeter [7] was proposed to address some of these issues.

However, the problem studied in this article relates only to the security issues associated with the generic scenario of Fig. 2. It is important to note that the real goal is to minimise the risk of fraud that might cost a fortune to providers and customers. The security requirements identified show the need for a specialised third-party, called Meter Inspection Authority (MIA), in whom providers and customers can trust. This proposed component offers specific services tailored to a metering scenario. The second most significant aspect of the security model is the set of protocols proposed. Despite their simplicity, they aim to fulfil the security requirements identified.

III. Security Requirements

An analysis of the business model (Fig. 1) raises a set of security requirements. Some of them are also discussed in [10] [11] [12] [13]. The classification used ranks them as follow: (1) **mandatory**: when the requirement must be fulfilled; (2) **highly desirable**: the requirement should be partially matched; and (3) **desirable**: the required may or not be fulfilled.

(A) Execution of safe code (meter system code,SLA/Tariff): Highly Desirable
The flexibility of the model allows the deployment of meters to any node in the network. Preferential positions might be either provider, customer or third-party provider's domains, as shown in Fig. 2. Therefore, this requirement assures the integrity of the system where the code is executed. A common threat to counter is the installation of any malicious code which may compromise the node.

(B) Matching of SLA/Tariff (high-level) and metering policyset (low-level): Mandatory
The meter policy set (configuration) must not be changed by any of the roles in the business model. When the policy set is prove to measure what the SLA/Tariff code states, this requirement will be matched. Any deviation from the original SLA/Tariff code after the translation process should be detected.

(C) Authentication when exchanging mobile codes, records and policysets: Mandatory
An authentication scheme between the roles is essential. A role should know beforehand who the other end of the communication is before beginning a data or control session. Thus, they must identify themselves using unique credentials. When reporting meter records, the meter must sign records systematically or on a random basis. In most common situations, the records will contain signatures of the meter code itself, SLA/Tariff code and SLA/Tariff translator.

(D) Levels of privacy and integrity: Mandatory
Meter records and policysets should be protected when stored locally or communicated over the network. A component of the EdgeMeter architecture, *the measurement manager*, must be able to modify such records. Customer and provider representative modules must only have read access to them. This requirement also brings the idea of authorisation schemes to the model. A role may be authorised to perform specific operations. The authorisation process takes place just after a role has identified itself through the authentication scheme.

(E) Avoiding cases of measurements repudiation: Desirable
No role may deny full or partial participation when exchanging data between each other, e.g., assuming that the measurements are taken by the customer. At some point, the meter installed in the customer's domain sends a report to the provider summarising the network usage. Later on, the provider notifies the customer that it has not received any meter data from him. What should the customer do in such a case?

(F) Denial of service (prevention and recovery) of the system: Desirable
Denial of Service (DoS) attacks in this context interfere with the normal usage or management of the meter system. Metering applications measure the usage of resources. As a result, the meter system may be considered a possible target of such attacks, especially when used with billing applications.
Such a problem is even worse in the cases of asymmetric benefits. It might be the case that a customer gets benefits from such DoS attack even if not intentionally. In this case, the provider loses revenue from whatever the attack is (e.g. a customer blocking the usage reports periodically sent to the provider).
On the other hand, the provider might get revenue out of this type of attack. For instance, suppose the meter responsible for taking measurements for the customer is flooded by packets. In such a case, the customer should not pay for this volume of background traffic. However, the provider will generate a bill for this spurious traffic.

IV. Security Model

This section presents simple security protocols derived from the Needham-Schroeder protocol [14] with added timestamps. These protocols address the issues outlined in the previous section.

Symbol	Description
KG	group key (shared)
KU_x	public key of x
KP_x	private key of x
$\{M_1, M_2, ..., M_r\}$	concatenation of messages $M_1, M_2, ..., M_r$
$\{M\}_K$	encryption of message M using key K
$\{M\}_{SIG_x}$	digital signature of x appended to message M
$S \longrightarrow R : \{M\}$	S sends message M to R

TABLE I
NOTATION

The notation shown in Table I will be used throughout this article in order to describe the protocols. Certification authorities (CA) are used to bind entities to public keys. Thus, extra protocol steps are required to assure that a given public key belongs to an entity. Such additional

(1) $MM \longrightarrow P : \{\{metersystem, version_1, t1_{MM}\}_{SIG_{MM}}, ID_{MM}\}_{KU_P}$

(2) $P \longrightarrow MIA : \{\{metersystem, version_1, SLA/Tariff translator, version_2, t1_P\}_{SIG_P}, ID_P\}_{KU_{MIA}}$

(3) $MIA \longrightarrow P : \{sealedcode, MIC_1, MIC_2, t1_{MIA}, ID_{MIA}\}_{KU_P}$

(4) $P \longrightarrow C/TPP : \{\{sealedcode, MIC_1, MIC_2, t2_P\}_{SIG_P}, ID_P\}_{KU_{C/TPP}}$

TABLE II
SECURITY PROTOCOL: INITIALISATION PHASE

steps are not shown explicitly and they are carried out before any of the protocol step proposed in this article takes place.

A. Meter Inspection Authority (MIA)

The Meter Inspection Authority (MIA), a third party, is used to establish trust between two entities. The main duties of the MIA cover the calibration and safety assurances of meters. Its services can be offered as a single or compound service to:

- Check whether the meter system code is safe. Occasionally, this task may be performed in real time by the execution environment (EE) used. In such cases, the code might be written in a type-safe language.
- Check whether the meter system is calibrated. This process verifies whether or not the code complies with a specification.
- Check whether the meter policy set can be safely installed in the meter system.
- Check for any deviation between the formal SLA/Tariff and meter policy set. The MIA can certify this property given that the output of the SLA/Tariff translator is consistent with the input.
- Issue Meter Inspection Certificates (MIC) in order to ensure that any of the above properties hold. Two types of MIC are available: (a) Safety certificate and (b) Calibration certificate.

B. Phase 1: Initialisation

During this phase, the *meter system* code is shipped out from the meter manufacturer to the points of measurements. These points may be one of the three possibilities discussed in Section II: provider, customer or third-party provider domains (Fig. 3).

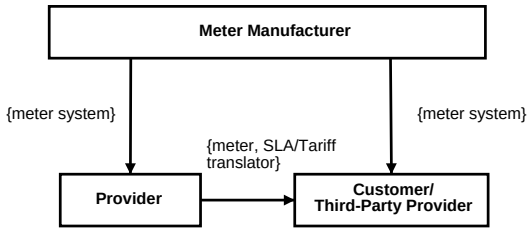

Fig. 3. Initialisation Phase

Fulfilling the security requirements associated with Fig. 3 is the intention of this section. There will be concerns regarding the execution of untrusted code (*meter system and SLA translator*). This applies especially in the cases where the code is shipped to the customer or third-party provider servers (shown on the right side of Fig. 3). In such a case, the requirement for the execution of safe code (Req. A) is strong.

There have been two fundamental approaches to fulfil this requirement for mobile code. The first, the language-based approach, constrains data type and memory access. Examples of such an approach include: Java [15], Safetynet [16] and Caml [17]. On the other hand, other schemes based on formal methods have been put forward. Proposals such as the Proof-Carrying code (PCC) [18] and Software Fault Isolation (SFI) [19] are included in this second approach.

The choice of a language-based approach was taken, in this article, to match the requirement of the safe execution of mobile code. Furthermore, authorisation schemes will be necessary for distributed meter management.

The integrity and privacy of the code shipped are also important issues to be addressed. How could it be guaranteed that the code, especially in a competitive market, was not modified or read by intruders? Thus, the use of asymmetric cryptography to sign the code in order to maintain the integrity is proposed.

One example showing the importance of guaranteeing privacy arises from the provider point of view. They might complain that their competitors took their meter code or SLA/Tariff in order to analyse how it works.

B.1 Protocol Description

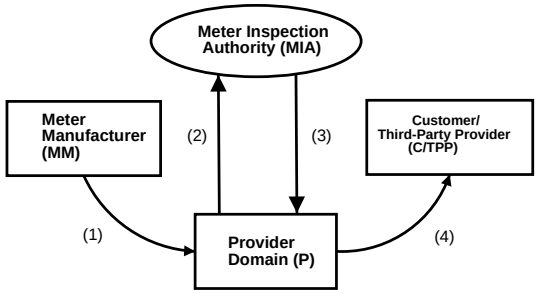

Fig. 4. Initialisation Phase:

The scenario of Fig. 4, which is a derivative of the diagram in Fig. 3, presents the provider buying a meter system to install in its customers/third-party providers' domain. Needless to say, this does not preclude customers and third-party providers from buying meter systems by themselves for verification purposes. This accounts for another scenario which is not detailed here due to space limitations.

Scenario A: *Providers buy the meter system from the Meter Manufacturer (MM). Then, the meter is installed at the customer/third-party provider's domain (Fig. 4).*

The Meter Manufacturer (MM) dispatches the message 1 containing the meter system to the Provider (P) (Table II). The message is signed by MM (SIG_{MM}) with an one-way hash function. Then, it is encrypted with P's public key (KU_P) to match the requirement of privacy. The timestamp *t1* informs the provider of the time the message was created. Distinct timestamp values will be used throughout the next steps for the purpose of message *freshness*.

During the second step shown in Table II, P sends a message containing the *meter system* and *SLA/Tariff translator* to MIA. A signature of P (SIG_P) is appended to the code which guarantees its authenticity. The message is encrypted using MIA's public key (KU_{MIA}).

In contrast, MIA analyses the code upon receiving the message from P. Then, it generates a certificate called MIC (Meter Inspection Certificate) with assurances. The certificate MIC_1 contains calibration and safety certification for the *meter system*. MIC_2 does the same for the *SLA/Tariff translator* but also adds extra assurance. The *SLA/Tariff*

(1) $SD \longrightarrow SR : \{\{SLA/Tariff, version\}_{SIG_{SD}}, t1_{SD}, ID_{SD}\}_{KU_{SR}}$

(2) $SR \longrightarrow MC : \{signedpolicyset, t1_{SR}, ID_{SR}\}$

(3) $MC \longrightarrow MS : \{signedmeterrecord, t1_{MC}, ID_{MC}\}KG_{MP,MC}$

(4) $MC \longrightarrow MP : \{signedmeterrecord, t2_{MC}, ID_{MC}\}KG_{MP,MC}$

TABLE III
SECURITY PROTOCOL: OPERATION PHASE

translator code not only has to prove to be safe and calibrated but also must contain assurance that its output (meter policy set) traces back to its input (SLA/Tariff). This is the requirement of SLA/Tariff and meter policy set matching (Req. B).

MIA creates a type of *meter seal* when signing the code (*meter system, SLA/Tariff translator*) with its signature (SIG_{MIA}). The seal guarantees that the code (*sealedcode*) will not be modified. By analogy, the electricity meters are also sealed to prevent anyone tampering with them.

During the last step, P sends the code sealed by the MIA to customers/third-party providers (C/TPP). They can check the certificates (*MIC1,MIC2*) and trust the code as conforming to a meter and SLA/Tariff translator specifications. This message sent from P is signed with P's signature for authentication and encrypted with C/TPP public key ($KU_{C/TPP}$).

C. Phase 2: Operation

At this stage, the meter system code should already be installed. Thus, the requirement of safe execution now applies to the mobile code carrying the SLA/Tariff structure. For that, the decision is to use a type-safe language with a sandbox inside the component *SLA/Tariff receptor (SR)* (Fig. 5).

The requirement of SLA/Tariff and metering policy set matching represents an interesting problem. How could one program written in a language A be traced back to its original source written in language B (equivalence problem)? The approach followed in this article adopts a third party that certifies the *SLA/Tariff translator*. It systematically tests the output (meter policyset) given an input (SLA/Tariff structure).

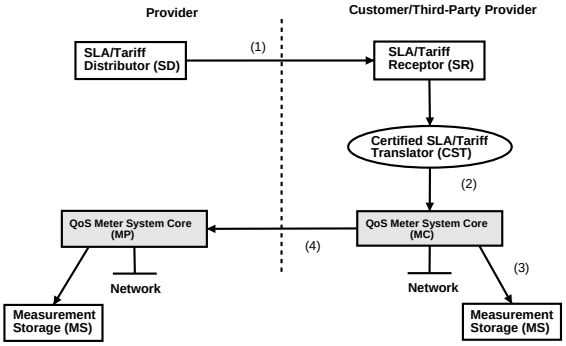

Fig. 5. Operation Phase

In Fig. 5, further security issues can be identified. The right levels of privacy and integrity are very desirable as meter records and policy sets should be protected.

When any party claims not to be getting the correct measurements, there will be a structure within the metering architecture to check this. First, all meter records will be signed by an internal component of the meter system, called measurement management. The records should be available only for reading through the group key (KG) explained in the protocol description below. Additionally, customer/third-party providers should not claim that the measurement does not match the actual usage since they have a certified and calibrated code from the MIA.

The provider should be able to set its own meter **on the fly** for auditing if it does not receive the meter records for reasons such as deletion of records from the measurement storage (MS), or network problems (congestion, traffic blocking). Doing so will account for new traffic and eventually lose a period of metering. Such a loss may be deemed irrelevant in most cases.

On the other hand, third-party providers/customers and providers may also install independent certified meters for auditing purposes. In this case, meter tampering can be prevented. However, the reconciliation, process of who has the correct measurement, should be done partially at legal instance.

C.1 Protocol Description

This section presents the protocol to secure the operation phase described by Fig. 5.

The protocol steps are presented assuming the notation below:

- **SD:** SLA/Tariff distributor
- **SR:** SLA/Tariff receptor
- **CST:** Certified *SLA/Tariff translator*
- **MC:** QoS Meter system code at customer/third-party provider
- **MP:** QoS Meter system code at provider
- **MIA:** QoS Meter inspection authority

Scenario A: *The SLA/Tariff translation is done at the customer/third-party provider's domain (Fig. 5). However, there will be situations where certain constraints may prevent having a SLA/Tariff translator being available. This latter scenario will not be explored further due to space constraints in this article.*

During the first step shown in Fig. 5 and Table III, the provider announces the SLA/Tariff to customers/third-party providers. In fact, such announcements may be performed in distinct ways. It could be used either multicast or unicast. Multicasting SLA/Tariff is used for the transmission of the same SLA/Tariff to groups of customers, whereas unicast is useful for communicating a specific SLA/Tariff to a customer. The SLA/Tariff is signed by the SLA/Tariff distributor (SD) and it is encrypted using the destination's public key (KU_{SR}) when using unicast. When multicast is used, the key for the group which the customer/third-party provider belongs to should be used.

The certified *SLA/Tariff translator* translates from SLA/Tariff to the meter policy set (step 2 of Fig. 5). The message 2 is not encrypted since the assumption is that local communication within the meter system takes place, therefore ruling out the confidentiality requirement for this particular case. However, the resulting meter policy set (*signed-policyset*) is signed by the *SLA/Tariff translator* using SIG_{CST} in an attempt to bind it to the original SLA/Tariff. This scheme may also be useful for the meter system when checking whether or not to accept the policy set, depending on where it comes from.

The third step shows where the meter record is stored (measurement storage module). This database stores data classified by SLA/Tariff and policy set. The message 3 is encrypted using the group key ($KG_{MP,MC}$). This enables the provider and customer meter system

to read the signed meter record (*signedmeterrecord*). However, the meter record is signed by an internal module of the meter system core, shown as 'MC' in Fig. 5.The signature of this inner component, called *measurement manager*, guarantees that any modification of the original meter record can be detected.

Finally, the meter system, installed in the customer/third-party provider domain (MC), starts to push out records to the meter system at the provider (MP). To accomplish this, message 4 is encrypted using the group key KG shared by MP and MC ($KG_{MP,MC}$). The meter record is signed with the *measurement manager* signature. The issue of key lifetime due to revoking or compromised keys leads to a re-keying process. Although it is a very interesting topic, the re-keying process is not covered in this article and is considered a topic for future investigation.

V. Conclusion

The Internet metering scenario considered in this work allows different measurement points over the network. A configurable meter code is deployed in some of them. While this flexibility accounts for the key aspect of the scenario, it does introduce security issues. The relationship among customers and providers might give further incentive to frauds such as tampering with the meter.

In this article a security model was proposed to secure a distributed network QoS metering. The requirements identified showed the need for a specialised third-party, called Meter Inspection Authority (MIA), in whom providers and customers can trust. Furthermore, a set of simple security protocols based on the PKI (Public Key Infrastructure) was introduced as to match the requirements.

The prevention of certain types of DoS attacks is an open issue and it is considered in the future work's agenda.

Acknowledgements

The authors are grateful to Steve Hailes, Adam Greenhalgh and Tristan Henderson from UCL for their comments and discussions. Special thanks to Arnaud Jacquet, Kennedy Cheng and Bob Briscoe from British Telecom Labs, Martin Berger (Queen Mary, University of London) and Igor Sobrado (University of Oviedo, Spain) for their valuable feedback. Finally, thanks for the comments received by the anonymous reviewers.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "RFC 2475 - An Architecture for Differentiated Services," *IETF*, Dec. 1998.

[2] R. Braden, D. Clark, and S. Shenker, "RFC 1633 - Integrated Services in the Internet Architecture: an Overview," *IETF*, June 1994.

[3] Bob Briscoe, Mike Rizzo, Jerome Tassel, Kostas damianakis, and Nicolai Guba, "Lightweight Policing and Charging for Packet Networks," in *Third IEEE Conference on Open Architectures and Network Programming - OpenArch 2000*, Mar. 2000.

[4] R.J. Gibbens and F.P. Kelly, "Resource Pricing and the Evolution of Congestion Control," in *Automatica*, 1999, number 35.

[5] M3I Project, "M3I - Market Managed Multiservice Internet," URL: http://www.m3i.org/.

[6] Mike Rizzo, Bob Briscoe, Jerome Tassel, and Kostas damianakis, "A Dynamic Pricing Framework to Support a Scalable, Usage-based Charging Model for Packet-switched Networks," in *IWAN'99 Springer-Verlag*, July 1999.

[7] Marcelo Pias and Steve Wilbur, "EdgeMeter: Distributed Network Metering," in *IEEE Globecom 2001*, Nov. 2001.

[8] Ross Anderson, *Security Engineering: A guide to Building Dependable Distributed Systems*, John Wiley, 2001.

[9] J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-To-End Arguments in System Design," *ACM Transactions on Computer Systems*, vol. 2, no. 4, pp. 277–288, 1984.

[10] David M. Chess, "Security Issues in Mobile Code Systems," in *Mobile Agents and Security; G. Vigna (ed.)*. 1998, Springer-Verlag.

[11] Wayne Jansen and Tom Karygiannis, "Mobile Agent Security," Tech. Rep. Special Publication 800-19, National Institute of Standards and Technology (NIST), Aug. 1999.

[12] Shimshon Berkovits, Joshua D. Guttman, and Vipin Swarup, "Authentication for Mobile Agents," in *Mobile Agents and Security; G. Vigna (ed.)*. 1998, Springer-Verlag.

[13] Antonio Corradi, Rebecca Montanari, and Cesare Stefanelli, "Security Issues in Mobile Agent Technology," in *7th IEEE Workshop on Future Trends of Distributed Computing Systems*, 1999.

[14] R.M. Needham and M.D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," *Communications of the ACM*, vol. 21, no. 12, Dec. 1978.

[15] Bill Venners, *Inside the Java 1.2 Virtual Machine*, Osborne McGraw-Hill, Aug. 1999.

[16] Ian Wakeman, Alan Jeffrey, Rory Graves, and Tim Owen, "Designing a Programming Language for Active Networks," in *Submitted to HIPPARCH Special Issue on Network and ISDN Systems*, 1998.

[17] Xavier Leroy, "Objective Caml," in *INRIA. http://caml/inria.fr/ocaml*, 1997.

[18] George C. Necula and Peter Lee, "Safe, Untrusted Agents using Proof-Carrying Code," in *Mobile Agents and Security; G. Vigna (ed.)*. 1998, Springer-Verlag.

[19] Robert Wahbe, Steven Lucco, Thomas Anderson, and Susan Graham, "Efficient Software-based Fault Isolation," in *14th ACM Symposium on Operating System Principles*, 1993.