Securing a Bandwidth Broker Architecture

Ch. Bouras[1,2], K. Stamos[1,2]
[1]Research Academic Computer Technology Institute, Riga Feraiou 61, GR-26221 Patras, Greece
[2]Computer Engineering and Informatics Dept., Univ. of Patras, GR-26500 Patras, Greece
Tel:+30-2610-{960375, 960316}
Fax:+30-2610-{969016, 960358}
e-mail: {bouras, stamos}@cti.gr

Abstract – In this paper we discuss the topic of the security in the context of a Bandwidth Broker's operation based on the existing literature on this issue and new techniques and methods. We propose mechanisms to enhance the security of the communications during the Bandwidth Broker operation both within a single domain and across different domains. The message exchanges for the operation of a distributed Bandwidth Broker architecture are enumerated. The purpose is to examine the main security risks for a distributed Bandwidth Broker architecture operating in a real-world environment and address them using the PKI architecture mechanisms. We also present a proposal for identifying misbehaving flows that intends to be both simple and effective. We finally evaluate the performance impact that our solution incurs to the operation of the Bandwidth Broker and show that it offers an advantageous trade-off in most cases where security considerations exist.

Keywords: Bandwidth Broker, security, SSL, SLA.

1.0 Introduction

The Differentiated Services (DiffServ) framework [1] is one of the basic architectures that have been proposed for QoS provision in the Internet. Because the Internet consists of numerous network domains with each one acting as an autonomous system, just using the current DiffServ framework does not solve the problem of providing end-to-end QoS, since each domain may be incompatibly configured. One entity that has been proposed in order to overcome this problem and provide end-to-end QoS across network domains is the Bandwidth Broker.

A Bandwidth Broker [2] is an entity responsible for providing QoS within a network domain. The Bandwidth Broker manages the resources within the specific domain by controlling the network load and by accepting or rejecting bandwidth requests. A user within the domain that is willing to use an amount of the network resources between two nodes, has to send a request to the Bandwidth Broker. The Bandwidth Broker chooses to either accept or reject a request based on the network load, its admission control policy and the Service Level Agreement (SLA). The SLA is the service contract between the service provider and every customer that describes the service that both the costumer and the service provider have agreed upon. The decision to accept or reject a request is made by the admission control module. In the case that the requested resource is managed by multiple domains, the Bandwidth Broker is also responsible for the inter-domain communication with Bandwidth Brokers of adjacent domains. This procedure requires communication between adjacent Bandwidth Brokers and also a special agreement between the domains.

The information that is handled by the Bandwidth Brokers has to be carefully protected from malicious attacks, since it enforces contracts in the form of SLAs that translate to service guarantees and specific costs. The issue of security for the Bandwidth Broker's communications can be divided in two main parts: Inter-domain security and intra-domain security.

Inter-domain security deals with the communications between Bandwidth Brokers that manage neighbouring domains. The effort on this area has concentrated on securing protocols such as the Simple Inter-domain Bandwidth Broker Signaling SIBBS [3], [4] protocol, that deal with the Bandwidth Broker communication across domains.

The SIBBS protocol is proposed by the Internet2 community in order to implement the inter-domain communications of resource reservation between the Bandwidth Brokers. It exchanges two pairs of

messages for QoS configuration purposes, the Resource Allocation Request (RAR) / Resource Allocation Answer (RAA) messages to request for a service, and the CANCEL / ACK messages to terminate the requested service. The transmitted information is sensitive and therefore has to be protected against possible security compromises. In [5], the authors outline the main security threats that inter-domain Bandwidth Broker communication has to protect against, and explain how the Public Key Infrastructure (PKI) can be integrated in order to produce a secure SIBBS implementation.

Intra-domain security has to deal with the communication between the Policy Decision Point (PDP) that is the Bandwidth Broker, and the Policy Enforcement Points (PEPs) that are typically the network routers that are appropriately configured in order to enforce the Bandwidth Broker's decisions. Also, in the case of a distributed Bandwidth Broker implementation, a large amount of sensitive internal Bandwidth Broker information is likely to be transmitted over the network and is therefore vulnerable if not properly protected.

The authors in [6] have proposed an efficient algorithm for the Bandwidth Broker's admission control module, with the intent of achieving satisfactory utilization of the network resources without heavily impacting the Bandwidth Broker's performance. In [7] the architecture has been extended so that it can support a distributed Bandwidth Broker architecture. In the case of a distributed Bandwidth Broker operation, the messages exchanged between the remotely positioned Bandwidth Broker modules have also to be secured, since in that case there is also a fair amount of intra-domain Bandwidth Broker communication that exchanges sensitive information related to the management of the network resources in the domain managed by the Bandwidth Broker. In the rest of this paper, our aim is to show how the whole architecture, including communication between Bandwidth Brokers, communication between PDPs and PEPs and communication between the Bandwidth Broker modules can be protected from a series of attacks that intend to compromise the system's security.

2.0 The Security Problem

Dealing with sensitive information such as the network resources management has to increase the awareness of possible security problems. The Public Key Infrastructure model (PKI) has been developed in order to deal with a number of possible attacks and protect against security, privacy and authentication violations. It is generally understood as the set of policies and software that regulate or manipulate the use of certificates and of public and private keys. Usually asymmetric encryption is used, which is based on a public key that can be disclosed to anyone, and a private key that is known only to its holder.

Our discussion intends to identify the ways with which the Bandwidth Broker implementation can be guarded against the various types of attack. In general, network attacks can be summarized in 3 broad categories:

- Integrity attacks: The attacker tries to compromise the correctness, timeliness, authenticity or quality of the information exchanged.
- Confidentiality attacks: The attacks tries to disclose sensitive information that should normally only be accessible for authenticated parties.
- Availability attacks: The attack tries to make the service unavailable to legitimate users.

Furthermore, a robust implementation also has to be capable of recovering from situations that do not pose a direct security threat, but can nonetheless compromise the operation of the Bandwidth Broker module. Such cases are:

- Equipment / software malfunction: One or more of the Bandwidth Broker modules do not operate as expected and, for whatever reason, produce invalid, unexpected or simply erroneous results.
- Users' misbehavior: Users that do not follow the rules that have been mutually agreed upon, by for example violating the SLAs and attempting to increase their network resource usage at the expense of other users. These users have to be identified and disciplined according to the policies that have been set in place for each case.

3.0 Enhancing Security of a Distributed Bandwidth Broker architecture

3.1. Message Exchanges of the Distributed Architecture

In order to identify the possible security considerations that should be taken into account, we enumerate in this section the communication messages that are exchanged during the operation of the distributed Bandwidth Broker architecture that was presented in [7]. In that architecture we used a model of a central Bandwidth Broker that is typically an available powerful server, and deals with the admission of the requested flows as long as the admission algorithm computations does not exceed its capabilities. The central Bandwidth Broker has complete knowledge of the managed network and the location of the secondary Bandwidth Brokers. Each secondary Bandwidth Broker is assigned a mutually exclusive subset of the network nodes in its neighborhood and keeps information about the state of the relevant part of the network. As soon as the central Bandwidth Broker is not capable of meeting the computation thresholds, it allocates subsets of the admission requests to the secondary entities. The secondary Bandwidth Brokers use the same admission control algorithm as the central Bandwidth Broker, but the smaller subsets they handle allow them to reside on relatively inexpensive hardware.

There are three phases that have to be examined, with regards to the number and type of messages exchanged:

- **Phase 1**: The central Bandwidth Broker handles all admission requests and manages the domain by itself.
  Therefore, we have no communication with the secondary Bandwidth Brokers during this phase. The only messages exchanged are between the central Bandwidth Broker and the endpoints requesting reservations. Since the architecture does not require a specific method for communicating the requests to the bandwidth management authority, depending on the chosen interface for the request of reservations, this communication channel can be encrypted using the SSL protocol.

- **Phase 2**: When the computation load becomes large enough to exceed a predetermined threshold, the central Bandwidth Broker proceeds to Phase 2 and starts distributing some of the admission requests. In particular, it distributes the admission requests whose source and destination are both managed by the same secondary Bandwidth Broker (we name these local requests).
  In this case, 2 types of information have to be transmitted from the central Bandwidth Broker to the secondary Bandwidth Brokers, namely the routing information, so that the secondary Bandwidth Brokers have the knowledge of the links where resources will have to be reserved, and the requests themselves. Both types of information are vulnerable to security attacks. For example, if an attacker could properly alter the routing information, the secondary Bandwidth Broker with the erroneous routing information might not be able to reserve the requested resources, and therefore break the service guarantees that a legitimate user has been promised. Compromise of the requests is an even bigger security threat, since in that case the attacker can divert the network resources in an undesirable way.

- **Phase 3**: If the remaining load on the central Bandwidth Broker remains large (larger than a times the threshold, where a can be reasonably set to a value of 1.5, as explained in [7]), we move to Phase 3, where the central Bandwidth Broker iteratively distributes even more admission requests in chunks. Requests are distributed according to the length of their predicted path (therefore the central Bandwidth Broker will release the control of the requests with the longest paths, for which it is the most suitable of managing, at the very latest stages). The selection of the value of parameter a is closely associated to the topology of the specific network.
  In this phase an increasing number of requests will have to be transmitted to the secondary Bandwidth Brokers, in addition with routing information to secondary Bandwidth Brokers that transition either from Phase 1 to Phase 2, or from Phase 2 to Phase 3. The central Bandwidth Broker keeps a structure that enables it to know exactly what information each secondary

Bandwidth Broker has, enabling the central Bandwidth Broker to only transmit the minimum needed information each time. For example, a secondary Bandwidth Broker that makes a transition from Phase 1 to Phase 2 and then back to Phase 1, will not receive routing information if it goes over to Phase 2 again, since the central Bandwidth Broker is aware that the necessary routing information has already been transmitted. (If there has been a network topology change in the meantime, the central Bandwidth Broker will send the update before any other information). This approach offers a double benefit, both in terms of reduced network overhead, and in reducing the sensitive information that is transmitted in the network.

The security for messages exchanged between the Bandwidth Broker components and messages directed to the PEPs can be enforced using the PKI model with lightweight certificates that do not have a large impact on the communication overhead imposed on the network.

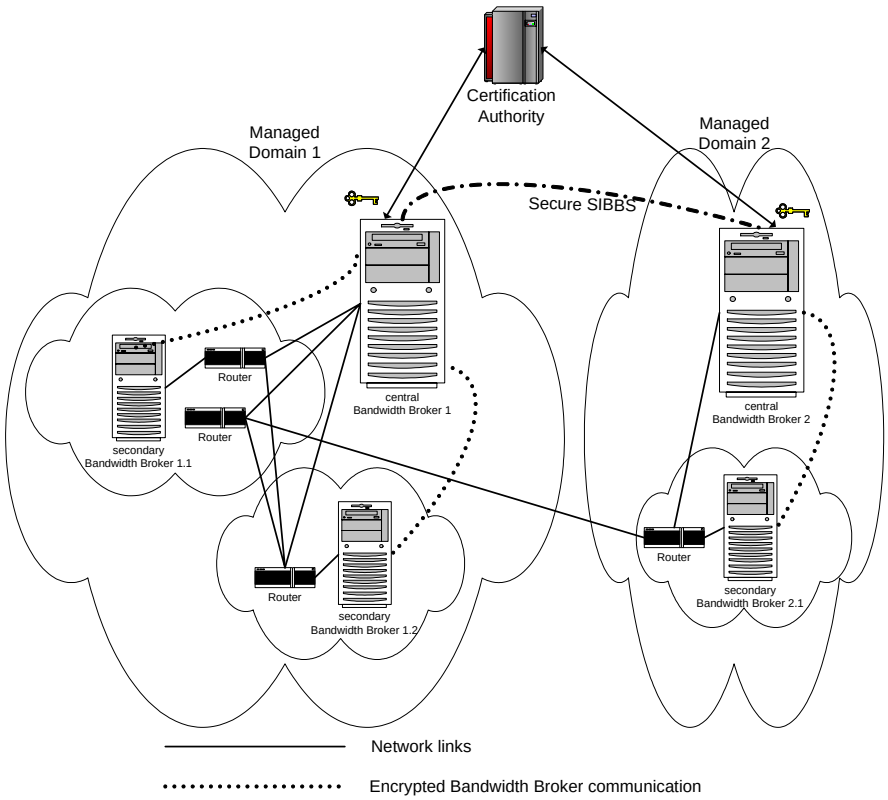

Figure 1: Security-enhanced distributed architecture

3.2. Protection against misbehaved modules

In order to make the architecture robust in the face of challenges such as the malfunction of a Bandwidth Broker component or its compromise by a malicious entity, we choose a two-folded approach:

- The central Bandwidth Broker, depending on its available computational resources, simulates a subset of the admission decisions that have been delegated to the secondary Bandwidth Brokers.
- Additionally, during Phase 3 the same non-local requests can be assigned to more than one secondary Bandwidth Brokers. More specifically, since in Phase 3 we can often have the situation where some of the secondary Bandwidth Brokers are inactive [7]. These modules can be activated and be assigned with sets of requests that would normally be handled by different secondary Bandwidth Brokers. This way, the robustness of the architecture is increased without any overhead (since these modules would otherwise be inactive). This approach leads to a correlation between the degree of achieved robustness and the amount of not harvested computational power across the architecture. The issue of increasing the robustness of the architecture in this regard, is addressed by

the increase in computational resources, which is a simpler and more easily understood and managed issue.

3.3. SIBBS Protocol – Inter domain Security

The SIBBS protocol proposed by Internet2 for inter-Bandwidth Broker communication needs to be enhanced with capabilities that provide authentication, integrity, and time sequence guarantee.
The authors in [5] have proposed a way to secure the SIBBS protocol for inter-Bandwidth Broker communication by incorporating the Public Key Infrastructure (PKI) model in the protocol operation. Their implementation offers authentication, integrity, timeliness and non-repudiation services based on the PKI technologies. Communication between Bandwidth Brokers is encrypted in order to protect against the types of attacks described in the previous chapter. An overview of the whole architecture described in the previous paragraphs can be seen in Figure 1.

4.0 Enforcing SLAs

Because the DiffServ model is based on an aggregate of flows, there is no inherent assurance that a misbehaving flow will not exceed its allocated resources, and therefore downgrade the service for legitimate users that share the same ingress point, leading to a serious problem for the provider of the guarantees agreed in the context of the SLA. It is an inherent advantage of the DifServ architecture, that this particular vulnerability is restrained at each ingress point, which is ultimately responsible for regulating its traffic. Also, depending on the marking mechanism, unauthorized endpoints may try to spoof the DSCP field that determines the behavior that a packet will receive from the QoS-enabled network. This vulnerability can be remedied by using some form of monitoring of the incoming traffic from a specific entry point. The distributed Bandwidth Broker components that were described in [7] can be utilized in order to perform such functions and police a specific part of the network.
In order for the monitoring mechanism to be scalable, each flow cannot be separately examined. A better approach, as proposed in [8], is to monitor aggregates of flows instead. These aggregates can be progressively reduced in size in the case that a misbehaving flow is recognized, until the illegal flow is identified. The problem with this approach is that a misbehaving flow can go unnoticed, if its excess usage is balanced by underutilization of the requested resources by other reservations.
An alternative approach is to implement a random sampling of flows and checking of whether there is any misbehavior. While this technique is simpler and therefore introduces smaller overhead, it can offer no strict guarantee on whether and when a misbehaving flow will be identified. It can however be combined with a feedback mechanism that retrieves information from the users and identifies problems or complaints, if such a mechanism is available.
Our proposal is to combine both approaches in an implementation that is able to utilize user feedback and random checking in order to identify misbehaving flows, while at the same time remaining scalable and lightweight by using the aggregation of flows.
The central entity of the Bandwidth Broker is assigned with an additional task of periodically monitoring the status of already accepted flows. It selects a set of flows and checks whether each flow is within the specified limits set by the relevant SLAs. Upon the reception of feedback from a user f_c, it adds flows that intersect at some point with f_c to set A, as can be seen in the following pseudocode:

```
// Select set A of flows:
focus = false
Upon reception of complaint by flow f_c
  for each link in f_c
    if resource usage exceeds proper limit at link l_f
      focus = true
      add to A flows that intersect with f_c at link l_f
    end if
  end for
if focus == false
  Add all flows that intersect with f_c to set A
```

```
// Binary search for violating flow:
while set A contains more than 1 flow
  break set A to separate sets A1,A2
  compare resource usage of flows in set A1 to SLA
  if resource usage for A1 > SLA, set A = A1
  else set A = A2
end while
```

The above algorithm checks each link that is used by the f_c flow to determine flows that are also utilizing the same network link. These are the most probable flows to be violating their SLA agreements, so the algorithm examines them first. This way, more efficiency is achieved in terms of quick response to SLA violations. Figure 2 shows an example of the process of selecting the set A. In this case, the total resource usage for Link 3 exceeds the allocated amount, and therefore all flows traversing Link3 are selected for forming set A on which the binary search algorithm is performed.

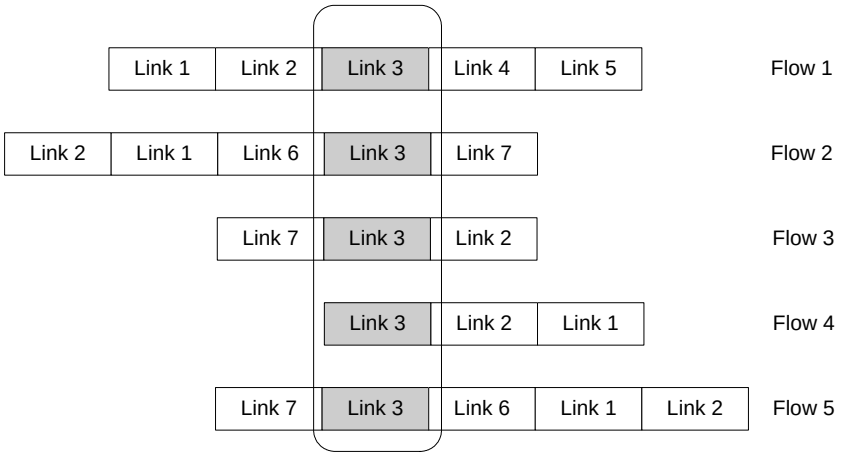

Figure 2: Selecting set A

At any time, an additional random checking of all flows is also performed. This means that even a flow that does not directly affect other flows (but nevertheless uses resources it has not been authorized to), will be successfully identified with a probability of

$$\frac{1}{F}(1+\frac{t_v}{t_c}) \quad (1)$$

where F is the total number of flows, t_v is the period of time for which a flow violates its SLA, and t_c is the time it takes the Bandwidth Broker to check whether a flow honours the SLA. This happens because in the time it takes for the Bandwidth Broker to check all flows (F t_c) a violating flow is vulnerable for time t_c+t_v , as can be seen in Figure 3. Greyed boxes are the cases where a violating flow will be caught, while white boxes are the cases where a flow that violates the SLA for time t_v will not be caught.

Compared to the monitoring of flows proposed in [8], our approach has the benefit that it can also discover breaches of the SLA even when they do not affect other users, but which are nevertheless lost potential revenue for the resource provider. If P_{affect} is the probability that a misbehaving flow will affect other users, then our approach compared to [8], has an increased probability of identifying the misbehaving flow at a factor of

$$(1\text{-}P_{affect})\ \frac{1}{F}(1+\frac{t_v}{t_c}) \quad (2)$$

since (1-P_{affect}) is the probability that the violating flow will not affect other users and our approach in that case has the probability of successful identification described in equation (1), while the approach in [8] has zero probability of successful identification in that case.

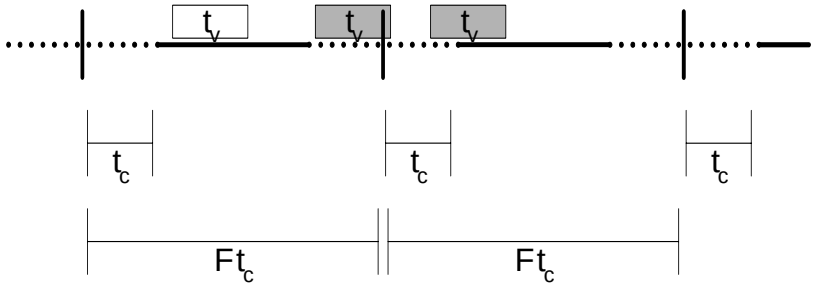

Figure 3: Flows violating SLA

Moreover, all misbehaving flows face the possibility of being identified, and the mechanism offers thereby an incentive for all flows to be well behaved. Perhaps most importantly, the mechanism starts to be effective even before a critical situation arises (such as a well behaved flow to be denied the service it has guaranteed). In terms of requirements, the mechanism is equally fast and lightweight since it introduces very small computational complexity.

5.0 Conclusions – Future Work

In this paper we presented the security considerations and the proposed solutions that are necessary for a Bandwidth Broker architecture that operates in an insecure environment. As the experience of the last decades with networks suggests, this has to be an important aspect of an architecture that intends to be widely deployed in large network topologies. Our work has focused both on securing the exchanged messages of the modules that comprise the architecture using the PKI infrastructure, and on protecting against compromised modules. Furthermore, we have examined how violations of SLAs can be effectively and inexpensively be identified.
Our future work will be focused on the real-world evaluation of the proposed architecture and on its detailed comparison with alternative architectures and security schemes.

6.0 References

1. S. Blake, D. Black, M. Carlson, M. Davies, Z. Wang, W. Weiss, “An Architecture for Differentiated Services”, lnternet RFC 2475, 1998
2. K. Nichols, V. Jackobson, L. Zhang, “A Two-bit Differentiated Services Architecture for the Internet”, lnternet RFC 2638, July 1999
3. QBone Signaling Design Team, Final Report, http://qos.internet2.edu/wg/documents-informational/20020709-chimento-etal-qbone-signaling/, July 9 2002
4. V. Sander, The security Environment of SIBBS, http://qbone.internet2.edu/bb/SIBBS-SEC.doc, June 2000
5. B. Lee, W.-K. Woo, C.-K. Yeo, T.-M. Lim, B.-H. Lim, Y. He, J. Song, “Secure Communications between Bandwidth Brokers”, Operating Systems Review 38(1): 43-57, 2004
6. C. Bouras, K. Stamos, “An Adaptive Admission Control Algorithm for Bandwidth Brokers”, 3rd IEEE International Symposium on Network Computing and Applications (NCA04), Cambridge, MA, USA, August 30 - September 1 2004, pp. 243 - 250
7. C. Bouras, K. Stamos, “Examining the Benefits of a Hybrid Distributed Architecture for Bandwidth Brokers”, The First IEEE International Workshop on Multimedia Systems and Networking (WMSN’05), to appear
8. S. Machiraju, M. Seshadri, I. Stoica “A Scalable and Robust Solution for Bandwidth Allocation”, Proceedings 10th International Workshop on Quality of Service (IWQoS), pp. 148-157, Miami Beach, FL, May 2002
9. V. Sander, W. Adamson, I. Foster, A. Roy, “End-to-End Provision of Policy Information for Network QoS”, Proceedings of the 10th International Symposium on High Performance Distributed Computing, pp. 115 – 126, 2001
10. Z. Zhang, Z. Duan, Y. Hou, “On Scalable Design of Bandwidth Brokers”, IEICE Transactions on Communications, Vol. E84-B, No.8, pp. 2011-2025, August 2001