Exploring the Security Requirements for Quality of Service in Combined Wired and Wireless Networks

Zubair Md. Fadlullah
Graduate School of
Information Sciences
Tohoku University, Japan.
zubair@it.ecei.tohoku.ac.jp

Tarik Taleb
NEC Europe Ltd.
talebtarik@ieee.org

Nidal Nasser
Department of Computing and
Information Science
University of Guelph, Canada.
nasser@cis.uoguelph.ca

Nei Kato
Graduate School of
Information Sciences
Tohoku University, Japan.
kato@it.ecei.tohoku.ac.jp

ABSTRACT

In the modern era of Internet, providing Quality of Service (QoS) is a challenging issue, particularly in resource-constrained wireless networks with delay-sensitive multimedia traffic. Real-time and multimedia services are now available to end-users over wired networks, Wireless Local Area Networks (WLANs), and Wireless Personal Area Networks (WPANs). While the usual trend is to provide the best possible QoS for these services, it is also imperative to deploy security requirements along with the QoS parameters. In this paper, we argue that the existing approaches for including security parameters (such as encryption/decryption key lengths) with QoS parameters (e.g., end-to-end delay requirements) lead to further security risks and consequently fail to provide an adequate solution. Through simulations, we point out the pitfalls of integrating delay and security support in the contemporary approaches. We also envision QoS^2, a framework integrating both quality of security and QoS, in order to provide possible solutions for solving these problems. We also demonstrate via simulation the effectiveness and strength of our adopted approach.

Categories and Subject Descriptors

C.2.0 [**Computer-communication networks**]: General - security and protection.

General Terms

Security.

Keywords

Encryption, Quality of Service, Quality of Protection.

1. INTRODUCTION

In order to provide Quality of Service (QoS) for real-time traffic and interactive multimedia applications, new architectural models such as Integrated Services (IntServ) [1] and Differentiated Services (DiffServ) [2] architectures have been widely used. While the field of QoS-oriented research has indeed matured over the years, recent developments in Wireless Personal Area Networks (WPANs), Wireless Local Area Networks (WLANs), and peer-to-peer networks have opened up new frontiers for QoS researchers. For example, due to the ever increasing use of Wi-Fi for transmitting QoS sensitive applications, which often contain sensitive and crucial information, it is of utmost importance to provide security along with QoS. Furthermore, depending on the natures of the applications used, the levels of security may be perceived differently by the end-users. While traditional QoS schemes may allow such applications to receive guarantees on particular QoS parameters such as bandwidth and delay, these schemes lack the support of differentiated security levels. Therefore, the notion of QoS must be extended to effectively accommodate multi-level security [3]. Some researchers have coined the term, Quality of Protection (QoP), which refers to the need of protection of sensitive information while maintaining QoS. QoP protects exchanges of information over wireless and wired media by employing end-to-end security mechanisms and cryptographic protocols. However, this may have sizable impacts on bandwidth and delay leading to QoS-degradation.

For providing QoS integrated with security for data intensive and time sensitive multimedia applications over IEEE 802.11-based wireless networks, Wenbo *et al.* [4] proposed a middleware adaptation scheme, to provide tunable end-to-end delay, which serves as a QoS parameter, along with variable security levels, i.e., QoP. Although their inspiring work is the first one in the field of assimilation of QoS parameters such as end-to-end delays with security attributes, it is not without its shortcomings. Through in depth analysis, we argue that under Denial of Service (DoS) like bandwidth consumption attacks [5], this scheme may indeed falter and become susceptible to sophisticated manipulations such as remote timing attacks [6]. Our work is not limited only with this problem formulation. We then focus on finding an adequate solution to address such security risks. Thus, we indicate the need for QoS^2, a framework that should take into consideration not just the end-to-end delay but also other QoS parameters along with various security attributes.

The rest of this paper is organized as follows. Section 2 surveys related research work in the field of Quality of Protection. With a comprehensive overview of the existing

IWCMC'09, June 21-24, 2009, Leipzig, Germany.

scheme [4], we elucidate its pitfalls and formulate the problem scope in Section 3. Simulation results are presented in Section 4 to illustrate the problems at hand. In Section 5, we envision QoS^2 and discuss, with giving empirical analyses, the possible course of action to solve these problems. Finally, Section 6 concludes the paper.

2. RELATED WORK

Security has been addressed rather implicitly in contrast with traditional QoS attributes such as latency, jitter, deadline, and fairness. The idea of Quality of Security Service (QoSS) [7] was conceived by assuming that the acceptable range of values for a security parameter relies on any of the three network modes, namely normal, impacted, and emergency modes. Each network mode is mapped into simple security level choices such as low, medium, and high levels. The security level is considered to be proportional to the strength of the employed cryptographic algorithm. For a particular network mode, QoSS computes the costs of determining the security levels, and presents a user with a list of possible security levels, from which the user can select the level that best suits his security-requirement. For computing these costs, QoSS considers factors such as disk space, CPU, memory, and available bandwidth. However, since it does not take into account end-to-end delay as a QoS parameter, it is not capable of providing end-to-end QoS support.

An online monitoring and self-protection mechanism for QoP is conceived in [8], which presents an Abnormality Distance (AD) metric based QoS scheme. This approach uses intrusion detection mechanisms to classify traffic streams into four categories, namely attacking flows, probable abnormal flows, probable normal flows, and normal flows. The AD values are employed to prioritize the routing of the packets belonging to these various flows. This approach reduces end-to-end delays even under DoS attacks. However, it does not consider cryptographic protocols and wireless topologies.

By integrating cross-layer security features in wireless LAN environments with IP mobility, Agarwal *et al.* [9] introduced a QoP model that takes into consideration authentication times, and also cryptographic overheads and throughputs. However, Agarwal *et al.* only considered encrypted protocols (such as 3DES and WEP-128) among the wireless hosts and corresponding access points. The end-to-end encrypted connections over networks consisting of both wired and wireless topologies have not been considered in this study.

In Security of Service (SoS) infrastructure [10], wireless users may state their security expectations during the negotiation for QoS in the Service Level Specifications (SLSs). Depending on the SoS parameters chosen by a user, four levels of security services can be selected, namely "high", "medium", "default", and "no" levels. These multiple levels of security services are provided by employing assorted algorithms with various parameters such as key lengths, block sizes, security protocols, and hash functions. Thus, SoS integrates security parameters within the SLSs to deploy differentiated security features for sensitive multimedia services.

The notion of secure QoS has also been introduced for ad-hoc wireless networks. While some researchers have focused on integrating QoP into ad-hoc routing protocols [11], others have directed their efforts towards QoS-aware authentication schemes for ad-hoc wireless networks [12]. Additionally, for 4G networks, Fu *et al.* [13] proposed an architecture called Seamless Mobility with Security and QoS (SeaSoS), which integrates QoS signaling with Authentication, Authorization, and Accounting (AAA) services. Apart from guaranteeing the QoS requirements of the users, SeaSoS achieves efficient authentication, authorization, and key exchange. However, upon authentication, it is also required to integrate security attributes with end-to-end QoS parameters. This issue actually has not been addressed in any of the aforementioned work.

A leading illustration of how security may be integrated as a dimension to existing QoS frameworks can be found in the middleware adaptation proposed by Wenbo *et al.* [4]. The users of IEEE 802.11-based wireless ad-hoc networks are presented with a set of security requirements and end-to-end QoS delay requirements. Depending on a user's chosen level of security and delay requirements, the middleware adaptor attempts to attain the minimum end-to-end delay while offering the user the highest possible security level, which is proportional to the encryption key-length. Thus, it achieves a balance between delay and security levels under varying network loads. Although this tunable QoS/QoP framework for QoS delay and security requirements serves as a pioneering work, it is not without its shortcomings. Especially, a bandwidth consuming attack may exploit the manner in which the encryption key-lengths are downgraded dynamically to maintain a reasonable end-to-end delay requirement for the user. The attacker may then launch remote timing like attacks [6] more effectively and quickly owing to the weakened encryption level. In this paper, we illustrate the significance of this problem of dynamically adjusting the lengths of the encryption keys with varying end-to-end delays. We also intend to devise adequate solutions for solving the same problem.

3. PROBLEM SCOPE: THE MULTI-LEVEL SECURITY MODEL AND ITS PITFALLS

In this section, we first briefly describe and explain the multi-level security model [4] for dynamically adjusting the security levels for varied QoS delay requirements of the users. We then point out the risks involving in this approach.

The end-to-end delays experienced by network packets are usually due to factors such as link delays and queueing at the intermediate nodes. In case of an end-to-end cryptographic protocol, the encryption and decryption processes at the application layers, as shown in Fig. 1, further contribute to the overall end-to-end delay, Δ, which can be expressed as:

$$\Delta = E + D + \sum_{i=1}^{n} T_i + \sum_{i=1}^{n-1} Q_i = E + D + T + Q \qquad (1)$$

where n denotes the number of segments along the path of propagation between the two end nodes U_1 and U_2. E, D, T, and Q indicate the delays due to encryption, decryption, transmission, and queueing, respectively. For multimedia applications in particular, a user (e.g., U_1 in Fig. 2) specifies a set of multiple levels of security requirements (s_{req}) and a set of acceptable QoS delay requirements (Δ_{req}) as follows:

$$s_{req} = \{s_{r_i} | s_{r_1} > s_{r_2} > \cdots > s_{r_i} > \cdots > s_{r_{min}}\} \qquad (2)$$

$$\Delta_{req} = \{\Delta_{r_i} | \Delta_{r_1} < \Delta_{r_2} < \cdots < \Delta_{r_i} < \cdots < \Delta_{r_{max}}\} \quad (3)$$

where s_{r_i} and Δ_{r_i} denote the multiple levels of security and the end-to-end delay requirements for various values of i, respectively. First, s_{r_i} indicates an encryption key length

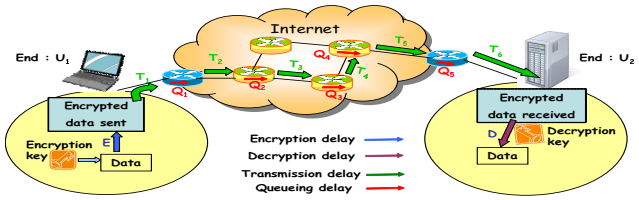


Figure 1: The overall end-to-end delay for an encrypted application.

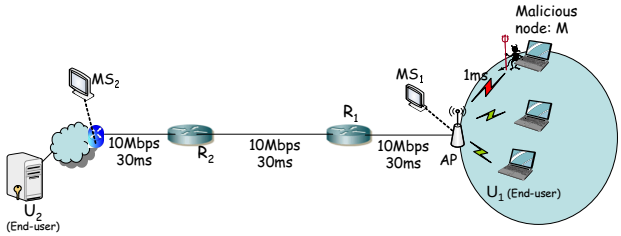


Figure 2: A Sample threat model.

and $s_{r_{i-1}}$ denotes a higher security level than s_{r_i}. Second, if the wireless network fails to guarantee the end-to-end delay requirement of Δ_{r_i}, the end-to-end delay requirement may be increased to $\Delta_{r_{i+1}}$. However, if the end-to-end delay exceeds $\Delta_{r_{max}}$, the maximum allowable limit specified in Δ_{r_i}, then that particular connection is dropped. By using a waiting time priority (WTP) scheduler, a middleware adaptor chooses a requirement pair (s_{req}, Δ_{req}) based on the observed end-to-end delay of the associated multimedia application, where s_{req} and Δ_{req} are selected from s_{r_i} and Δ_{r_i}, respectively. The details of the algorithms implemented in the middleware adaptation scheme can be found in [4]. One major flaw in the design of this middleware adaptation scheme is in its inability to take into consideration the role an attacker (M) may play, as depicted in Fig. 2. By sniffing traffic at the access point AP, M attempts to intercept sensitive information that flows between the two end users, U_1 and U_2.

For instance, M can launch remote timing attacks [6] to extract the private key from U_1, which can be used to decipher the transmitted information between U_1 and U_2. As long as the traffic is encrypted with a significantly large cryptographic key, it is both difficult and time-consuming for M to decipher and use the information passed between U_1 and U_2. In order to compromise the middleware adaptation scheme that tunes (s_{req}, Δ_{req}), M then attempts to increase Δ between U_1 and U_2. A simple way of achieving this is to generate a large volume of traffic along the path between U_1 and U_2 (e.g., UDP flooding). For instance, in Fig. 2, M overwhelms the link (AP-U_2) by means of bandwidth consumption attacks [5]. A bandwidth consumption attack is a type of DoS that uses up all or a major portion of available bandwidth on the target link. Such an attack eventually causes Δ between U_1 and U_2 to increase, which will prompt the middleware scheduler to lower the s_{req} as a trade-off to the increased Δ_{req}. Consequently, the encryption level weakens to such an extent that M can indeed mount successful remote timing attacks within a reasonably short time. Thus, another aspect of QoP, namely key recovery by traffic analysis, is observed in the analysis here. It should be noted here that although most properly designed and protected schemes may thwart some of these attacks, systems failing to safeguard against these risks may become seriously vulnerable or compromised.

4. EVALUATION OF THE THREAT MODEL

4.1 Simulation Set-up

By using Network Simulator (NS-2) [14], a simple topology, as depicted in Fig. 2, is set up. The wireless user U_1 and the wired node U_2 use VoIP applications, simulated over UDP traffic. The size of data packets for these applications is set to 1000 bytes. In our simulation, we consider encryption and decryption delays for encrypting and decrypting these packets based on the average encryption/decryption delays measured in the OpenSSL package of SuSE Linux 10.3. Other simulation parameters are shown in Table 1.

In contrast to the delay-security pairs in [4] that takes no statistical history into account, we adopt a profiling scheme based upon the "usage history" of the end hosts in order to obtain the QoS delay and security levels to be offered to U_1. When U_1 establishes a connection with U_2 for the first time, it sends a UDP echo request to U_2. As U_2 receives this echo request, it sends a response packet back to U_1. By computing the Round Trip Time (RTT) for this echo request-response operation, the average end-to-end delay between U_1 and U_2 can be estimated as follows:

$$\Delta_{avg} = \frac{1}{2} \cdot RTT \tag{4}$$

However, Δ_{avg} does not provide an accurate information about the individual end-to-end delays along the uplink and downlink directions of U_1 and U_2. To circumvent this problem, U_2 inserts into a response packet the time at which the echo request was received, and then sends it back to U_1. After receiving this response packet, U_1 can compute the end-to-end delays along both its uplink and downlink directions. For simplicity, let us consider only the uplink direction for U_1, along which the end-to-end delays between U_1 and U_2, denoted by Δ_{up} values, are recorded at the beginning of each VoIP session between the two ends over a long period of time. The lowest Δ_{up} value, Δ_{min}, which was about 110ms, is considered to correspond to the least congested network scenario. Δ_{min} is used in Eq. 5 to construct baseline requirements of end-to-end delay Δ_{r_i}; where i=1, 2,..., l, and l represents the number of delay requirement levels (l=4 in our simulation):

$$\Delta_{r_i} = \epsilon \cdot \Delta_{min}; (1 + (i-1) \cdot \mu) \leq \epsilon \leq (1 + i \cdot \mu) \tag{5}$$

where μ is set to a small value of 0.09 since this is the best choice to establish four delay intervals of equal lengths in the range of 110ms to 150ms (maximum latency for VoIP traffic is 150ms). This implies that the first level of end-to-end delay requirement is upper-bounded by 109% of Δ_{min}. Similarly, the second level of delay requirement is within [109%-118%) times that of Δ_{min}. Thus, for demonstration and testing, we formulate the end-to-end delay levels offered to U_1 for this simulation, as shown in Table 2. $\Delta_{r_{max}}$ is set to 150ms, the upper bound of level 4, which is the lowest delay requirement level (i.e., the most relaxed one) perceived by U_1.

We maintain four security levels (in terms of cryptographic key lengths of 512, 256, 192, and 128-bits, respectively). The average delays contributed by encryption and decryption operations at the two ends (U_1 and U_2) are enlisted in Table 3. The system offers U_1 the currently available levels of delay

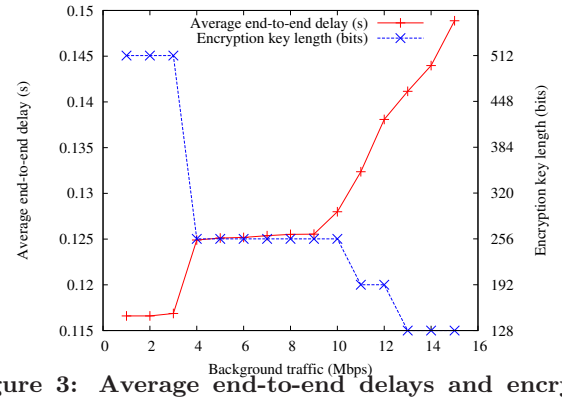


Figure 3: Average end-to-end delays and encryption key lengths for various background traffic-rates (Mbps).

and security. U_1 chooses one or more levels for both security and delay. The system tries to ensure the highest level of delay and security chosen by U_1 whenever possible. Otherwise, it degrades the security level to keep the delay within the highest level chosen by U_1. If the system cannot ensure the highest level of delay for the lowest level of security chosen by U_1, it relaxes the delay requirement by switching to the subsequent delay level. The simulation results obtained in all experiments have a 95% confidence level with 5% confidence intervals.

4.2 Results

The VoIP traffic between U_1 and U_2, data rates of which are set between 64 to 100 Kbps, are considered legitimate in these simulations. Initially, a malicious user M introduces low background traffic of just one Mbps. Such a moderate attack rate does not hamper the average end-to-end-delay of the packets belonging to the legitimate traffic, and consequently U_1 and U_2 can maintain relatively large keys (with size of 512-bits) for encrypting/decrypting the multimedia information. The background traffic is then varied from one to 15 Mbps, and the corresponding average of the upper bounds of the end-to-end delays and encryption key lengths are plotted in Fig. 3. Up to background traffic rates of 3 Mbps, U_1 manages to sustain this high level of security in terms of large cryptographic key sizes of 512-bits. As M continues to generate more malicious traffic gradually, the overall background traffic rate reaches 4 Mbps causing the average end-to-end delay to increase and exceed the "delay-security requirement" for U_1. Consequently, the security

Table 1: Simulation Parameters.

Simulation Parameter	Value
Wireless Parameters	
Propagation model	Two Ray Ground
MAC	802.11 a
Antenna	Omni directional
Routing	Infrastructure-based
Data rate supported	54 Mbps
Other Parameters	
Simulation time	20 s
End-to-end application	VoIP over UDP
Packet size	1000 bytes
Cryptographic algorithm	RSA (key sizes: 128-512 bits)

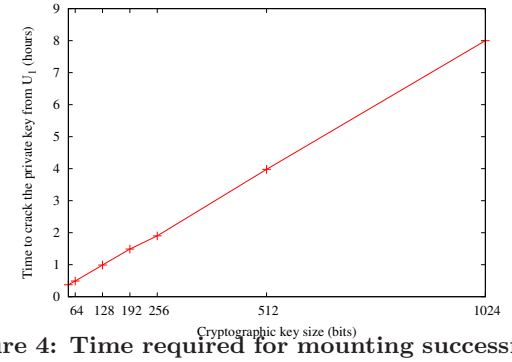


Figure 4: Time required for mounting successful remote timing attacks by M against U_1 for varying sizes of cryptographic keys.

level is downgraded to the next level by employing smaller cryptographic key sizes of 256-bits. This "delay-security requirement" is satisfied for the background traffic rates up to 10 Mbps. Similarly, for attack rates of 11-12 Mbps, the security level is further degraded. As the background traffic continues to rise even more and reaches 13 Mbps, the security level is brought down to the lowest grade (i.e., 128-bit cryptographic keys) since the average end-to-end delay increases substantially and exceeds 140ms.

By using a series of sophisticated remote timing attacks as demonstrated in [6] against U_1, the attacker M can then exploit this situation. Based on the attack parameters in [6], we plot the time required for successfully carrying out such attacks against varying sizes of cryptographic keys at U_1 in Fig. 4. The empirical results demonstrate that the time required to compromise the key decreases linearly, and is indeed low (below an hour) for key lengths of 128-bits. Therefore, it is essential to consider the stage in the network where the attack traffic rate reaches 11 Mbps as the "critical point", during which the targeted link capacity is saturated and the security is seriously compromised.

5. ENVISIONED SOLUTION: QOS^2

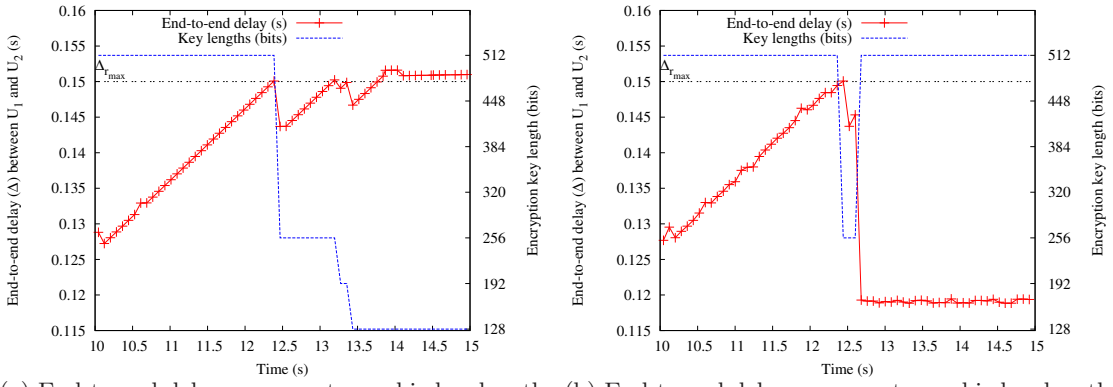
We argue here that there are two ways of securing QoS, namely efficient traffic filtering and QoS scheduling mechanisms. The former approach aims at providing bandwidth, delay, and security guarantees to legitimate flows. The latter needs to be designed to provide QoS assurances for various kinds of applications and to maintain fairness among various flows based on their priorities while still meeting their security requirements and also achieving high bandwidth utilization. In this paper, we only focus on the former strategy, i.e., an effective filtering mechanism to prevent bandwidth consumption attacks from consuming network resources to maintain QoS delay-security requirements.

To this end, we need to detect such bandwidth consumption attacks [5] mounted by malicious users (e.g., M in

Table 2: End-to-end delay requirement levels of U_1 and U_2.

Level (i)	End-to-end delay, Δ_{r_i} (ms)
1	[110, 120)
2	[120, 130)
3	[130, 140)
4	[140, 150)

Notation: $[\ a,b\ ) = \{\Delta_{r_i} \mid a \leq \Delta_{r_i} < b\}$.



(a) End-to-end delays vs. cryptographic key lengths over time using the non-QoS^2 method.

(b) End-to-end delays vs. cryptographic key lengths over time using the QoS^2 method.

Figure 5: Comparison of the end-to-end delays (between U_1 and U_2) and cryptographic key sizes without and with the proposed scheme during the bandwidth consumption attack.

Fig. 2). To mitigate such a threat, we derive inspiration from the approach adopted by Katabi *et al.* [15], which however only detects malicious attacks at the server and does not deal with bandwidth consumption attacks at all. Our envisioned approach, that we refer to as QoS^2, to deal with bandwidth consumption attacks, is described as follows. QoS^2 adopts traffic sniffing entities called Monitoring Stubs (MSs) (introduced in our earlier work [16] [17]), which are placed at strategic points in the considered network, such as aside core routers and gateways. The MSs are employed to avoid additional computational overheads at the routers. In our approach, AP allows M to connect to U_2 (or other intended destination). MS_1 then commences to monitor for an imbalance between the incoming flow from M destined for U_2 and the corresponding outgoing flow along the direction from U_2 to M. In a connection-oriented protocol such as Transfer Control Protocol (TCP), the number of packets to and from a source is usally evenly matched. Even in the case of real-time interactive multimedia-based applications based on unresponsive protocols such as UDP, there is a balanced bi-directional flow [5]. The bandwidth consumption attacks are crafted in such a manner that the victim is unable to reply to all the incoming packets resulting in an imbalance in the packet flow rates between the victim and the attacker. In Fig. 2, to detect such an imbalance, we use MSs (MS_1 in this case) to monitor the flows into a router/access point (e.g., AP) from a source (e.g., the flow from M to U_2 or U_2's network) and also its corresponding outbound flow (i.e., the flow from U_2's network destined for M), data rates of which are denoted by D_{in} and D_{out}, respectively. Let L_{avg} be the average packet size associated with the incoming flow. A time interval δ_j is set to the minimum half-path RTT value associated with the source and the destination in the concerned flow. Then we estimate the last n incoming packets to monitor after every δ_j as follows:

$$n = \frac{D_{in}}{L_{avg}} \cdot \delta_j; \qquad j = 1, 2, 3, ... \tag{6}$$

After at least n packets have arrived at AP from M, MS_1 starts monitoring within each δ_j the time instants at which the initial and the n^{th} of the last n packets arrive, denoted by t_{1_j} and t_{n_j}, respectively. The number of packets p_j in the corresponding outgoing flow within the period from t_{1_j} to t_{n_j} is also recorded. Let (r_{δ_j}) be the ratio between n and p_j. An imbalanced bi-directional flow due to a bandwidth consumption attack is characterized by $(r_{\delta_j} >> 1)$, which causes MS_1 to increment the entry of a Counting Bloom Filter, CBF. To prioritize a particular flow, an Anomaly Score (AS) value that ranges from zero to three is assigned to the flow. In this case, the incoming flow at the router that contributes to an increment at CBF as evaluated by MS_1, is assigned a high AS value (e.g., 2) to classify it as a probable abnormal traffic. Its corresponding outbound flow is assigned the lowest AS value (zero). This is done because the upstream traffic from U_2 may cause an imbalance in M's WLAN. But since this upstream traffic is in response to the initial requests from M, it is considered to be a part of an already established flow, and so the upstream traffic is considered legitimate. On the other hand, for a balanced bi-directional flow, r_{δ_j} should be close to one. In case of such a balanced flow, the corresponding entry in CBF is decremented by one. When the counter for this entry exceeds a threshold β (which is set to 16 since four bit counters are employed in CBF), the source is considered to be malicious and its IP address is, therefore, blacklisted. MS_1 then instructs AP to drop further packets associated with that incoming flow, thereby protecting the link.

A simulation is conducted over the previously described simulation topology in Section 4, to evaluate the effectiveness of the proposed approach. At the 10^{th} second of the simulation, M starts to launch the bandwidth attack along the link $(M - AP - U_2)$ at a high rate of 15 Mbps and with an average packet size of 3000 bytes. δ_j was set to 92ms, the minimum half-path RTT value between M and U_2. By using Eq. 6, n is computed to be 58. In this case, the chosen delay requirement level of U_1 is Δ_{r_4} (i.e., the most relaxed delay level as shown in Table 2). In addition, U_1 chooses all

Table 3: Average delays for performing cryptographic operations at end hosts for different cryptographic key sizes.

Cryptographic key size	Contributed delay (ms)
512	24
256	16
192	14
128	10

four available key lengths. The end-to-end delays (between U_1 and U_2) and the lengths of the employed cryptographic keys are plotted over time in Figs. 5(a) and 5(b) for the traditional middleware approach [4] and the proposed QoS^2 method, respectively. As evident from these results, under the non-QoS^2 approach, the end-to-end delay increases considerably with time. This compels the security levels to be gradually downgraded to keep the end-to-end delay within the acceptable range of ($\Delta_{r_{max}} = 150$ms). As demonstrated in Fig. 5(a), U_1 and U_2 maintain the highest security level (with key sizes of 512 bits) up to 12.4^{th} second, after which the system switches to the next level (i.e., with keys of 256 bits) to mitigate further end-to-end delays that may exceed $\Delta_{r_{max}}$. The security levels are further reduced over time, and eventually the middleware approach adopts the lowest security level at 13.4^{th} second and continues to use the same. Even with this lowest security level, the end-to-end delay exceeds $\Delta_{r_{max}}$ at 13.8^{th} second, and as a result legitimate packets are dropped from that point. On the other hand, MS_1 in the proposed QoS^2 approach monitors the flow imbalance and detects the bandwidth consumption attack along the affected link at 12.68^{th} second, i.e., in 2.68s since the start of the attack, as shown in Fig. 5(b). At this point, M's IP is blacklisted and further packets in the attack flow are dropped. Thus, we prevent the $(AP-U_2)$ link from being overwhelmed by M. By using the proposed method, the end-to-end delay in the affected link then drops substantially which allows U_1 and U_2 to switch back to stronger cryptographic keys of 512 bits. Thus, the end-to-end delay requirements are maintained and high security is guaranteed in the proposed QoS^2 approach.

Thus, we envision the Quality of Service with security framework, which we refer to as QoS^2, to formulate preventive measures once an attacked link is found. It should be worth mentioning that the adopted filtering method is designed in such a way that it can complement the basic admission control mechanisms. Admission control schemes assign acceptable QoS to aggressive clients (both legitimiate and illegitimate ones) when the server workload is low and their perceived QoS is degraded with increasing traffic. Aided by an effective filtering approach like QoS^2, such admission control methods do not need to consider illegitmiate clients or attackers, which exhibit aggressive demands for resources.

6. CONCLUSION

In this paper, we have investigated the existing multi-level security model that attempts to relate security requirements with the QoS framework by naively degrading cryptographic key-lengths to adjust increasing end-to-end delays. By establishing a threat model, we have demonstrated via simulations the vulnerability of this approach under bandwidth-consuming attacks. We have envisioned a mechanism that takes such attacks into consideration and protects the differentiated security approach. Our work also demonstrates clearly the need to have QoS^2, a robust QoP framework that will integrate various QoS attributes with different security parameters.

7. REFERENCES

[1] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview," IETF, RFC 1633, Jun. 1994.

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

[3] S. N. Foley, S. Bistarelli, B. O'Sullivan, J. Herbert, and G. Swart, "Multilevel Security and Quality of Protection," in *First Workshop on Quality of Protection,* Como, Italy, Sep. 2005.

[4] W. He and K. Nahrstedt, "An Integrated Solution to Delay and Security Support in Wireless Networks," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Las Vegas, NV, USA, Apr. 2006.

[5] S. Abdelsayed, D. Glimsholt, C. Leckie, S. Ryan, and S. Shami, "An Efficient Filter for Denial-of-Service Bandwidth Attacks," in *Proc. IEEE Globecom*, San Francisco, USA, Dec. 2003.

[6] D. Brumley and D. Boneh, "Remote Timing Attacks are Practical," in *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol. 48, No. 5, Aug. 2005, pp. 701-716.

[7] C. Irvine, T. Levin, E. Spyropoulou, and B. Allen, "Security as a Dimension of Quality of Service in Active Service Environments," in *Proc. 3 rd Annual International Workshop on Active Middleware Services*, San Francisco, CA, USA, Aug. 2001.

[8] S. Hariri, G. Qu, R. Modukuri, H. Chen, and M. Yousif, "Quality-of-protection (QoP) - an Online Monitoring and Self-protection Mechanism," in *IEEE J. on Selected Areas in Communications*, Vol. 23, No. 10, Oct. 2005, pp. 1983-1993.

[9] A. K. Agarwal and W. Wang, "On the Impact of Quality of Protection in Wireless Local Area Networks with IP Mobility," in *J. Mobile Networks and Applications*, Vol. 12, No. 1, Jan. 2007, pp. 93-110.

[10] S. Duflos, B. Kervella, and V. C. Gay, "Considering Security and Quality of Service in SLS to Improve Policy-based Management of Multimedia Services," in *Proc. 6 th International Conference on Networking*, Sainte-Luce, Martinique, France, Apr. 2007.

[11] S. Yi, P. Naldurg, and R. Kravets, "Integrating Quality of Protection into Ad Hoc Routing Protocols," in *Proc. 6 th World Multi-Conference on Systemics, Cybernetics, and Informatics*, Orlando, FL, USA, Aug. 2002.

[12] P. Muppala, J. Thomas, and A. Abraham, "QoS-based Authentication Scheme for Ad Hoc Wireless Networks," in *Proc. International Conference on Information Technology: Coding and Computing*, Las Vegas, NV, USA, Apr. 2005.

[13] X. Fu, D. Hogrefe, S. Narayanan, and R. Soltwisch, "QoS and Security in 4G Networks," in *Proc. 1 st Annual Global Mobile Congress*, Shanghai, China, Oct. 2004.

[14] The Network Simulator - NS-2, available at: http://www.isi.edu/nsnam/ns/

[15] S. Kandula, D. Katabi, M. Jacob, and A. Burger, "Botz-4-Sale: Surviving DDoS Attacks that Mimic Flash Crowds," in *Proc. Usenix 2nd Symposium on Networked Systems Design and Implementation*, Boston, USA, Jul. 2005.

[16] Z. M. Fadlullah, T. Taleb, N. Ansari, K. Hashimoto, Y. Miyake, Y. Nemoto, and N. Kato, "Combating Against Attacks on Encrypted Protocols," in *Proc. IEEE ICC*, Glasgow, Scotland, Jun. 2007.

[17] T. Taleb, Z. M. Fadlullah, K. Hashimoto, Y. Nemoto, and N. Kato, "Tracing back Attacks Against Encrypted Protocols," in *Proc. ACM IWCMC*, Honolulu, Hawaii, USA, Aug. 2007.