Data Protection Based on Physical Separation: Concepts and Application Scenarios

Stefan Lindskog[1], Karl-Johan Grinnemo[2], and Anna Brunstrom[1]

[1] Department of Computer Science,
Karlstad University, SE-651 88 Karlstad, Sweden
{Stefan.Lindskog, Anna.Brunstrom}@kau.se
[2] TietoEnator AB,
Lagergrens gata 2, SE-651 15 Karlstad, Sweden
Karl-Johan.Grinnemo@tietoenator.com

Abstract. Data protection is an increasingly important issue in today's communication networks. Traditional solutions for protecting data when transferred over a network are almost exclusively based on cryptography. As a complement, we propose the use of multiple physically separate paths to accomplish data protection. A general concept for providing physical separation of data streams together with a threat model is presented. The main target is delay-sensitive applications such as telephony signaling, live TV, and radio broadcasts that require only lightweight security. The threat considered is malicious interception of network transfers through so-called eavesdropping attacks. Application scenarios and techniques to provide physically separate paths are discussed.

1 Introduction

In the last few years, we have experienced a steadily growing interest in using the Internet as a vehicle for e-banking, e-commerce, virtual company networks, telephony, live TV, and other applications requiring secure communication. To this end, network security has become pivotal for the future of the Internet and Internet-based solutions.

Currently, network security is almost exclusively accomplished through encryption. For example, e-banking and e-commerce typically take place over Secure Sockets Layer (SSL) [9] or Transport Layer Security (TLS) [6] connections. However, although encryption gives adequate protection, it may lead to severely degraded network performance in terms of latency and throughput. Specifically, Apostolopoulos et al. [2] demonstrated a throughput reduction of more than 90 % when TLS (using RC4 and MD5) was used to access Netscape and Apache Web servers as compared to no encryption at all. Furthermore, Burke et al. [3] showed that applications running on high-end microprocessors are not even likely to saturate a T3 (approximately 45 Mbps) line. To this end, various selective encryption schemes [12, 19, 23, 24] that produce less overhead compared to ordinary encryption schemes, such as RC4, DES, and AES, have recently been

O. Gervasi et al. (Eds.): ICCSA 2005, LNCS 3483, pp. 1331–1340, 2005.

proposed. The basic idea of selective encryption is to offer lightweight security by encrypting only a subset of the data. Such schemes are intended to be used when the computational overhead produced by encryption and/or decryption must be reduced and a less stringent security level is acceptable.

Using selective encryption is, however, only one way to accomplish lightweight security. Furthermore, as pointed out by Rushby and Randell [22], the basis for security is separation. Thus another, in a sense more straightforward, way to obtain lightweight security is to physically separate the data to be protected, i.e., to partition and send the data along different routes. Physical separation has previously been used in other contexts. For example, Deswarte et al. [5] proposed the use of physical separation to accomplish intrusion tolerance in a distributed file archiving system, i.e., to make the file archiving system resilient against single-point attacks. However, to our knowledge, physical separation has not been used to obtain transmission security in networks. To this end, this paper suggests using physical separation as a complement to encryption for delay-sensitive network applications that require only lightweight security. It is our belief that, similar to selective encryption, physical separation could be an alternative for applications such as telephony signaling, live TV, and other multimedia applications. In other words, it could be an alternative for applications where latency and throughput requirements outweigh the importance of absolute protection against malicious (eavesdropping) attacks. According to Pfleeger and Pfleeger [18], providing adequate protection is a key security principle [1]. The principle implies that data should be protected to a degree that is consistent with their value.

The remainder of the paper is organized as follows. Section 2 presents our idea of providing data protection through physical separation. The threat model is discussed in Section 3. Section 4 focuses on application scenarios and on how to provide multiple physically separate paths in reality. Finally, Section 5 concludes the paper with some final remarks and a few words on future work.

2 Data Protection Through Physical Separation

Security is typically implemented through one or more security services. In particular, a combination of protective and detective services is often used. Defense diversity is achieved by combining two or more security services. Diversity of defense is a general security principle used to enhance security [4]. Firewalls are used for example to block suspicious network traffic to and from internal networks, and intrusion detection systems (IDSs) are used as a complement to firewalls to detect insider and outsider intrusion attempts as well as successful intrusions. However, neither firewalls nor IDSs are suitable security tools for protecting data that are transferred over an insecure network, such as the Internet. Instead, some form of data protection service is needed. The type of service necessary depends on what is to be protected.

[1] The same principle is in [17] referred to as the *principle of timeliness*.

Data protection services are used to achieve data confidentiality, data integrity, data authenticity, and/or non-repudiation [25]. A data confidentiality service ensures that transmitted data are accessible only for reading by authorized parties, while a data integrity service ensures that only authorized parties are allowed to modify transmitted data. A data authenticity service ensures that the origin and/or the source of data are correctly identified. Finally, a non-repudiation service ensures that neither the sender nor the receiver of a message can deny the transmission.

Data protection for network transfers has traditionally been implemented exclusively through cryptographic separation, and various cryptographic systems are widely used today. The major disadvantage of cryptographic separation, however, is that it requires adequate computational resources (at least) at the endpoints. For this reason, we propose the use of physical separation as a complement to existing cryptography-based data protection services for applications with lightweight security requirements. Cryptography and physical separation may also be combined to achieve protection diversity.

The idea of physical separation is to simultaneously send messages belonging to the same data stream on multiple physical paths. Protection is thus provided by the geographical fragmentation and scattering of data. Figure 1 shows two communicating hosts, A and B. In this case, host A is the sender and host B the receiver. Both hosts are multi-homed and equipped with two physical network interfaces, A_1 and A_2, and B_1 and B_2, respectively.

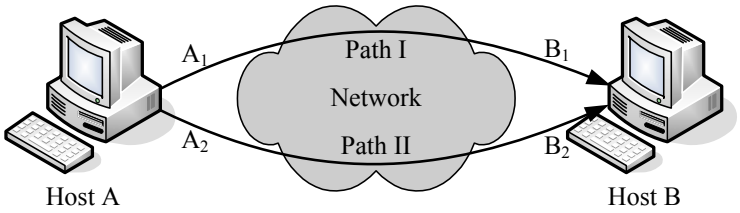

Fig. 1. Physical separation of two communication paths

Two distinct paths can be obtained, Path I and Path II, by using the different network interfaces, which is illustrated in Fig. 1. These two paths are then used simultaneously during transmission of data. Specifically, when connecting a multi-homed host to the network, each network interface is bound to a separate IP address. An application running on a multi-homed host can thus select the network interface on which to send data. Different strategies for distributing data on these paths can be used depending on the application. An even distribution of data on the different paths is suitable in some situations, while, in other situations, it is not. For example, when multimedia data are transferred, the amount of traffic sent on the different paths may need to be adjusted with respect to path characteristics such as bandwidth, delay, jitter, etc. The distribution may also be affected by security considerations. Imagine, for example, a map with a secret target marked with an "X". Neither the map nor the mark is sensitive by

itself but together they are. Increased security may thus be achieved by simply sending the map and the "X" mark on two physically separate paths.

3 Threat Model

The specific threat considered in this paper is malicious interception of network traffic through so-called eavesdropping attacks. Eavesdropping attacks are a serious threat to the confidentiality (or secrecy) of data transferred in a network. The person conducting such an attack is referred to as an eavesdropper. In this paper, our hypothesis is that an eavesdropper needs to acquire access to all or at least most of the data sent over the different paths to successfully perform an eavesdropping attack. This means that an attacker must identify all used paths, gain access to the traffic, and finally decode the data.

An eavesdropping attack on a particular victim is most easily performed near the victim's physical network connection. If the wire that connects the victim's computer is accessible, a protocol analyzer can be used to intercept all traffic that passes through the wire. Another option is to use a so-called network sniffer. When a network sniffer is used in a broadcast network, such as a non-switched Ethernet network or an IEEE 802.11 wireless network, the network interface card (NIC) on the computer executing the sniffer is configured in promiscuous mode. Promiscuous mode implies that all traffic sent over the broadcast network is intercepted by the NIC and forwarded to the sniffer for further processing.

Interception of traffic to and/or from a particular user is much more complicated if the point for eavesdropping is many hops away from the end nodes. In the core network, traffic from a large set of users will pass. Thus, the data processing necessary to filter out the relevant traffic will require a great deal of computational resources. In addition, traffic to and/or from the victim may be routed on different paths from session to session and can also be re-routed within a session due to network failures etc. Furthermore, if proxies or network address translators (NATs) are used, it might not be evident who is actually communicating.

When data on multiple communication paths are fragmented and scattered, eavesdropping attacks become much more difficult. The highest degree of protection through physical separation is achieved when data are routed on fully distinct communication paths all the way from the sender to the receiver. However, in many IP-based network architectures, fully distinct paths can not easily be guaranteed. Still, since the endpoints are the most susceptible parts to eavesdropping attacks, we believe that it is satisfactory in many real situations to guarantee physical separation of the traffic closest to the endpoints.

4 Application Scenarios

It is evident that physical separation as a mean to achieve security is more applicable in some situations than others. In the following, we give some examples of application scenarios in which physical separation may be used to provide

lightweight data protection and outline how physical separation can be realized in these scenarios.

4.1 SS7 over IP

Logically, the Public Switched Telephone Network (PSTN) comprises two networks: one call-traffic network for the transmission of speech, data, fax etc., and one signaling network for the transfer of control or signaling information. Specifically, the signaling network enables transfer of control information in between nodes in connection with: traffic control procedures such as call set-up, supervision, and release; services such as toll-free (800/888) and toll (900) wireline services, and roaming in wireless cellular networks; and, finally, network management procedures such as blocking and de-blocking of trunks.

The predominant signaling system used in today's PSTN is Signaling System #7 (SS7) and, with VoIP emerging as a competitive alternative to the PSTN, it has become increasingly important to enable interworking between SS7 and IP. To this end, the IETF Signaling Transport (SIGTRAN) working group has defined a framework architecture, the SIGTRAN architecture [15], that describes how to transport SS7 signaling information over an IP-based network. The SIGTRAN architecture defines a set of adaptation protocols that insulate the upper layers of the SS7 protocol stack from IP and thus make it possible to run SS7, essentially unaltered, over an IP-based network. Furthermore, the SIGTRAN architecture defines a new transport protocol, the Stream Control Transmission Protocol (SCTP) [26].

SCTP is the common denominator for all SIGTRAN networks and, although SCTP is now considered a general transport protocol on a par with TCP and UDP, it is still true that SCTP evolved from the SIGTRAN working group and the requirements of SS7 signaling. Similar to TCP, SCTP is a connection-oriented, reliable transport protocol offering ordered (sequential) delivery of user messages. However, contrary to TCP, SCTP is message oriented and provides support for multi-streaming and multi-homing. Multi-streaming enables SCTP to send separate transactions as independent units and thereby avoid so-called head-of-line blocking, which happens when a lost message in one transaction effectively blocks one or several other transactions.

Support for multi-homing was included in SCTP to provide quick failure detection and recovery, which is particularly important for SS7 signaling. Network-level timers running in today's SS7 networks monitor message delivery performance and may generate alarms that trigger message re-routing if they expire as a result of extraordinary delays. Thus, this network monitoring functionality must extend to VoIP networks to maintain end-to-end, carrier-grade telecommunications quality.

Figure 2 illustrates how the SCTP multi-homing support works. SEP-A and SEP-B are two dual-homed signaling endpoints. Each combination of source and destination IP addresses, e.g., (P_1,P_2), (A_1,A_2), (P_1,A_2) etc., constitutes a network path. One network path is always selected as the primary path, and (P_1,P_2) functions as the primary path in Fig. 2. Provided the primary path is available, all packets are sent on this path. The remaining paths function only

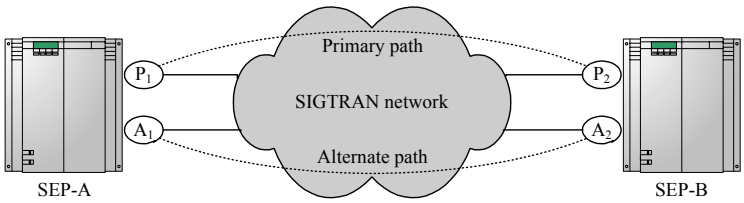

Fig. 2. Multi-homing support in SCTP

as backup or alternate paths. However, if the primary path becomes unavailable, one of the alternate paths takes over as the primary path. Thus the SCTP multi-homing support does not include load-balancing.

Although multi-homing in SCTP and SIGTRAN is exclusively intended for failure recovery, we believe that multi-homing could also be used to accomplish physical separation, and thus enhance security, for delay-sensitive signaling traffic. Specifically, we consider physical separation through the use of SCTP multi-homing a viable complement to IPSec [11], the security mechanism recommended for SIGTRAN by IETF—especially, since IPSec has been shown [13] to decrease the throughput of small packets, such as signaling packets, by approximately 60 % as compared to unencrypted transfer.

Since path switching in SCTP normally only takes place during failure recovery, physical separation must rely on some other mechanism than the SCTP failover mechanism. One possible solution would be to employ the SCTP changeover procedure, which provides for explicit changes of primary paths. However, the SCTP changeover procedure is primarily intended for infrequent path switching, and it might thus be necessary to improve its performance in order for it to be useful for physical separation. Another solution would be to consider one of the SCTP implementations that support load sharing, e.g., LS-SCTP [8] and RivuS [20].

To actually accomplish physical separation, it must be guaranteed that the primary and alternate paths are kept distinct from each other or, at least, only overlap in backbone routers and/or links. While this is not possible in the general case, with paths crossing several autonomous systems, it is indeed possible in controlled, dedicated SIGTRAN networks. As a matter of fact, since distinct paths also improve path redundancy and failure recovery, several telecommunication operators already plan to build their SIGTRAN networks with distinct primary and alternate paths between signaling endpoints.

Two techniques that have been proposed to establish distinct paths are redundant networks and MultiProtocol Label Switching (MPLS) [21]. Redundant networks are fairly straightforward and simply entail having the signaling endpoints connected to two or more distinct networks. However, to our knowledge this technique has not yet been deployed in any real SIGTRAN network. Thus it remains to be shown that the costs of building several separate networks are

actually compensated for in terms of reliability, availability, and faster failure recovery.

While it is too early to completely rule out the idea of using redundant networks, MPLS is currently the most compelling technique for establishing distinct paths. An MPLS network or Internet consists of a set of nodes, called Label Switched Routers (LSRs), which are capable of switching and routing packets on the basis of a label that has been appended to each packet. Labels define a flow of packets between two endpoints or, in the case of multicast, between a source endpoint and a multicast group of destination endpoints. For each distinct flow, called a Forwarding Equivalence Class (FEC), a specific path through the network of LSRs is defined. Thus MPLS is a connection-oriented technology. Associated with each FEC is a traffic characterization that defines the QoS requirements for that flow. The LSRs do not need to examine or process the IP header, but rather simply forward each packet on the basis of its label value. The forwarding process is therefore simpler than with a traditional IP router. Furthermore, it is envisioned that label-switched paths could be determined, and perhaps modeled, with traffic engineering tools that reside on network management workstations.

4.2 Application-Controlled Multi-homing

In the previous section, multi-homing was handled at the transport layer. Another alternative is to directly control multi-homing at the application layer, or possibly in a supporting middleware. This could be suitable for multimedia applications, for example, or more generally for applications where the transmitted data essentially comprise several independent, or almost independent, parallel data streams.

The main advantage of controlling multi-homing directly in the application, as compared to controlling it from lower layers such as the transport layer, is that it gives greater control over how data are transmitted. It also has the advantage of not being dependent on a specific transport protocol. Furthermore, from a security perspective, the use of parallel connections adds some level of extra protection. An eavesdropper that has managed to collect all data over all paths still needs to identify the transport layer connections that carry data that belong to the application of interest. Furthermore, once the connections have been identified, it still remains for the eavesdropper to appropriately reassemble the data.

Application-controlled multi-homing is, however, not without problems. For one thing, it entails that the reassembly of packets from different paths must be taken care of by the application itself. Second, the problem of enforcing distinct paths remains, provided of course that the paths have to be completely separated in the first place. Physical separation at the endpoints can be achieved by simply sending traffic over several network interfaces, possibly using different network technologies and/or media. For example, some parts of the traffic could go over wireline paths while other parts are sent via wireless access lines. If greater control of the physical paths is required, MPLS [21] could be used in the same

way as discussed in Section 4.1, and thus provide explicit label-switched paths between endpoints. However, application-controlled multi-homing also opens the way for application specific routing strategies through the use of so-called overlay networks [16].

An overlay network can be viewed as a logical network implemented on top of a real physical network. Each node in the overlay also exists in the underlying physical network. However from the viewpoint of the overlay, the nodes are not only capable of routing packets on the basis of their destination address, but are also able to process and forward packets in application-specific ways. Thus, an overlay network could assist a multi-homed endpoint in enforcing distinct paths. For example, a technique similar to the one proposed for Resilient Overlay Networks (RONs) [1] could be used. In a way similar to a RON, nodes could reside in a variety of routing domains and cooperate with each other to forward data. Since routing domains or Autonomous Systems (ASs) rarely share interior links, flows routed through different ASs are likely, but not guaranteed, to be forwarded on distinct paths.

Another technique for enforcing distinct paths when using overlay networks may be to employ a routing underlay network as suggested by Nakao et al. [14]. They propose a new architectural element, a routing underlay, that sits between an overlay network and the underlying IP network. The overlay network queries the routing underlay when it makes routing decisions. The routing underlay in turn extracts and aggregates topology information from the underlying IP network and answers the queries of the overlay network. Specifically, one of the services that could be offered to the overlay by the underlay network is to return a distinct path between an originating and destination endpoint. In fact, this is one of three services that are actually proposed in [14].

Finally, it can be mentioned that recent developments in routing technologies make possible new alternatives, apart from MPLS and overlay routing, to obtain distinct paths even though the paths cross several ASs. Examples of this include route controllers, such as the Peer Director [7], which assists BGP in routing traffic through multi-homed routing domains (domains connected to several ISPs), and the BANANAS framework [10], which permits source-based multipath inter-domain routing.

5 Conclusions

In this paper, we propose the use of physical separation as a complement to encryption for delay-sensitive applications that require only lightweight security. A threat model is presented. Further some application scenarios are highlighted, and alternative solutions for providing separate paths in these scenarios are discussed.

Since establishing physically distinct paths is a non-trivial task in IP networks, a large part of the description of application scenarios addresses this issue. Transport layer controlled multi-homing using SCTP in conjunction with either redundant networks or MPLS are suggested as feasible solutions. MPLS

could also be used together with application-controlled multi-homing. In this context, application-specific routing through the use of overlay networks is another possible solution.

As a next step we intend to make a feasibility study of the use of SCTP for transportation of SS7 signaling traffic on physically separate paths. The purpose in particular is to evaluate the complexity in terms of reassembly of data and to study the consequences of simultaneously sending data on multiple paths with regard to the SCTP congestion control mechanisms.

Acknowledgments

This research is supported in part by grants from the Knowledge Foundation of Sweden and from the CMIT research platform at Karlstad University in Sweden. The second author would also like to thank TietoEnator AB for their support of his research.

References

1. D. Andersen, H. Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient overlay networks. In *Proceedings of the 18th ACM Symposium on Operating System Principles (SOSP 2001)*, pages 131–145, Chateau Lake Louise, Canada, October 2001.
2. G. Apostolopoulos, V. Peris, and D. Saha. Transport layer security: How much does it really cost? In *Proceedings of the Conference on Computer Communications (IEEE INFOCOM)*, volume 2, pages 717–725, New York, New York, USA, March 1999.
3. J. Burke, J. McDonald, and T. Austin. Architectural support for fast symmetric cryptography. *ACM SIGOPS Operating Systems Review*, 34(5):178–189, December 2000.
4. D. B. Chapman and E. D. Zwicky. *Building Internet Firewalls*. O'Reilly & Associates, 1995.
5. Y. Deswarte, L. Blain, J. C. Fabre, and J. M. Pons. Security. In D. Powell, editor, *Delta-4: A Generic Architecture for Dependable Distributed Computing*, chapter 13, pages 329–339. Springer-Verlag, 1991.
6. T. Dierks and C. Allen. RFC 2246: The TLS protocol version 1.0, January 1999.
7. Radware: Peer Director. http://www.radware.com/content/products/pd, January 2, 2005.
8. A. A. El Al, T. Saadawi, and L. Myung. LS-SCTP: A bandwidth aggregation technique for stream control transmission protocol. *Computer Communications*, 27(10):1012–1024, June 2004.
9. A. Frier, P. Karlton, and P. Kocher. The SSL 3.0 protocol. Netscape Communication Corporation, November 1996.
10. H. Tahilramani Kaur, S. Kalyanaraman, A. Weiss, S. Kanwar, and A. Gandhi. BANANAS: An evolutionary framework for explicit and multipath routing in the Internet. In *Proceedings of the ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA 2003)*, pages 277–288, Karlsruhe, Germany, 2003.
11. S. Kent and R. Atkinson. RFC 2401: Security architecture for the Internet protocol, November 1998.

12. S. Lindskog, J. Strandbergh, M. Hackman, and E. Jonsson. A content-independent scalable encryption model. In *Proceedings of the 2004 International Conference on Computational Science and its Applications (ICCSA'04), part I*, pages 821–830, Assisi, Italy, May 14–17, 2004.
13. S. Miltchev, S. Ioannidis, and A. D. Keromytis. A study of the relative costs of network security protocols. In *Proceedings of the FREENIX Track: 2002 USENIX Annual Technical Conference*, pages 41–48, Monterey, California, USA, June 2002.
14. A. Nakao, L. Peterson, and A. Bavier. A routing underlay for overlay networks. In *Proceedings of the ACM SIGCOMM 2003)*, pages 11–18, Karlsruhe, Germany, August 2003.
15. L. Ong, I. Rytina, M. Garcia, H. Schwarzbauer, L. Coene, H. Lin, I. Juhasz, M. Holdrege, and C. Sharp. RFC 2719: Framework architecture for signaling transport, October 1999.
16. L. Peterson, T. Anderson, D. Culler, and T. Roscoe. A blueprint for introducing disruptive technology into the Internet. In *Proceedings of the First ACM Workshop on Hot Topics in Networking (HotNets 2002)*, Princeton, New Jersey, USA, October 2002.
17. C. P. Pfleeger. *Security in Computing*. Prentice-Hall, 2nd edition, 1997.
18. C. P. Pfleeger and S. Lawrence Pfleeger. *Security in Computing*. Prentice-Hall, 3rd edition, 2003.
19. M. Podesser, H. P. Schmidt, and A. Uhl. Selective bitplane encryption for secure transmission of image data in mobile environments. In *Proceedings of the 5th IEEE Nordic Signal Processing Symposium (NORSIG'02)*, Tromsø/Trondheim, Norway, October 2002.
20. RivuS project homepage. http://sourceforge.net/projects/rivus/, January 2, 2005.
21. E. Rosen, A. Viswanathan, and R. Callon. RFC 3031: Multiprotocol label switching architecture, January 2001.
22. J. M. Rushby and B. Randell. A distributed secure system. In *Proceedings of the 1983 IEEE Symposium on Security and Privacy*, pages 127–135, Oakland, California, USA, April 1983.
23. A. Servetti and J. C. De Martin. Perception-based selective encryption of G.729 speech. In *Proceedings of the 2002 IEEE Internatinal Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 621–624, Orlando, Florida, USA, May 2002.
24. G. A. Spanos and T. B. Maples. Performance study of a selective encryption scheme for security of networked, real-time video. In *Proceedings of the 4th International Conference on Computer Communications and Networks (ICCCN'95)*, pages 72–78, Las Vegas, Nevada, USA, September 1995.
25. W. Stallings. *Cryptography and Network Security: Priniples and Practice*. Prentice-Hall, 2nd edition, 1998.
26. R. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. J. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. RFC 2960: Stream control transmission protocol, October 2000.