A Design Methodology for HW/SW Security Protocols

Alberto Ferrante

The ability to communicate has become of fundamental importance for every activity of human life: companies need their branches to be constantly connected and to communicate with their customers and partners; human beings need to communicate with each other, with companies, and institutions.

Security and privacy are also an important need of the modern world: the rising competitiveness among industries imposes an increasing level of protection for each company's confidential information; private information of individuals also need being protected or, anyway, people should be enabled to reveal pieces of information to other people of their choice.

This need for security is obviously in contrast with the need for communication. As a matter of fact, sending information over any communication mean could expose them to possible eavesdroppers. The only way to solve this contrast is to introduce some security mechanisms in communications. Communication security is crucial for economic and social development. Many security mechanisms have been studied and deployed over the years. Presently used mechanisms are based over cryptography or, speaking of few top advanced applications, quantum cryptography. The latter technology is the future for a limited number of applications as it requires peculiar technological conditions. Traditional cryptographic techniques will probably continue to be used for common applications for many years to come. Both traditional and quantum cryptographic algorithms need to be included into suitable communication protocols to be utilized effectively. As a matter of fact, cryptographic algorithms need mechanisms for exchanging keys between the parties that are involved and for managing the secure connections. These are some of the services provided by secure protocols.

Specialized hardware is often required to support high network performances when secure protocols are considered: often general purpose CPUs cannot provide the necessary computational capacity. Gilder's and Moore's laws say that this situation is not going to improve with time: while Moore's law says that computational capacity is doubling every 18 months, Gilder's one says that available network bandwidth doubles every 12 months.

Goal of the work presented in this dissertation is to study a comprehensive design methodology for mixed hardware/software architectures dedicated to security protocols. The IPSec (IP Secure) protocol suite is taken as a reference for this work, as it has assumed great importance, being also included as the mandatory security mechanism in the new version of the IP protocol, IPv6. IPSec is a suite of protocols, which provides encryption and/or authentication services for the IP datagrams. The main contributions of this thesis is summarized in the following paragraphs.

A profiling of one of the current IPSec software implementations has been performed: this allowed to understand in which cases hardware acceleration is really necessary and the performance requirements of this hardware. Some guidelines for IPSec configuration have also been derived by this work. An abstract UML model of IPSec has also been developed: this model has been used as a reference during the design phase and for developing a testing methodology for IPSec-based systems. Test patterns are generated by means of a coverage criterion for the statecharts of the model. These statecharts are describing the internal states of the protocols.

The presently used systems have been studied and some relevant aspects of them have been optimized: three different scheduling algorithms for IPSec packet processing have been developed. These algorithms allow distributing the cryptography-related processing of IPSec packets among multiple cryptographic accelerators and a software implementation of the same algorithms. The first algorithm proposed tries to minimize the waiting time of each of the packets (i.e. the time packets wait before being processed). The second algorithm is based on the first one and it addresses the performance problems caused by small packets (40% of the

datagrams circulating on the Internet are 40-byte long) as present systems usually show poor performance in this case. The third algorithm also tries to minimize the waiting time for the packets, but it also provides Quality of Service support.

The need to overcome the performance problems of the present IPSec implementations lead to the development of an innovative architecture for a System on Chip (SoC). This SoC allows to efficiently process IPSec traffic and it is based on the *flow-through* idea: all the network traffic, even the packets not requiring any IPSec processing, flows through the SoC before arriving to the host system. This way, IPSec packets do not require being sent back and forth to the accelerators, thus allowing for a faster processing. This SoC needs to be able to process the IP traffic efficiently. Therefore, its internal architecture required careful designing. The number of data transfers inside the chip were minimized by adopting a shared memory model. Thus, the speed of the chip is mostly limited by the memory speed: its bandwidth should roughly be eight times the one that we want to support for the network. Beyond the shared memory, the main functional units of the SoC are: an inbound I/O management unit and an outbound I/O management unit, three IPSec database management units (one for the inbound, one for the outbound, and one for the IKE databases), a scheduling unit, one or more cryptographic cores, and an unit for header processing. The IKE functionalities (i.e., secure connection negotiation) can be either supported internally by a dedicated unit, or externally by a suitable combination of hardware and software. Some schemes for interconnecting more than one of these SoCs in parallel have also been proposed to support bandwidth requirements exceeding the ones provided by a single chip. The architecture of two functional units, namely the IKE and the database management ones, have been developed and simulated.

Bibliography

[1] Antonio Vincenzo Taddeo, Alberto Ferrante, and Vincenzo Piuri, “Scheduling Small Packets in IPSec-based Systems," in CCNC 2006. Las Vegas, NV, USA: IEEE, 8 Jan. 2006.

[2] Alberto Ferrante, Vincenzo Piuri, and Fabien Castanier, “A QoS-enabled Packet Scheduling Algorithm for IPSec Multi-accelerator Based Systems." in Computing Frontiers. Ischia, Italy: ACM, May 2005, pp. 221-229.

[3] Alberto Ferrante, Vincenzo Piuri, and Jeff Owen, “IPSec Hardware Resource Requirements Evaluation," in NGI 2005. Rome, Italy: EuroNGI, 18 Apr. 2005.

[4] Alberto Ferrante, Giuseppe Piscopo, and Stefano Scaldaferri, “Application-driven Optimization of VLIW Architectures: a Hardware-Software Approach," in Real-Time and Embedded Technology Applications. San Francisco, CA, USA: IEEE Computer Society, 7 Mar. 2005, pp. 128-137.

[5] Alberto Ferrante, Uljiana Boiko, Antonietta Lo Duca, and Vincenzo Piuri, “A Testing Methodology for IPSec-based Systems," in SoftCOM 2004. Dubrovnik and Split, Croatia, Venice, Italy: University of Split, Oct. 2004, pp. 22-26, sponsored by the IEEE Communication Society and by the UNESCO Communication and Information Society.

[6] Fabien Castanier, Alberto Ferrante, and Vincenzo Piuri, “A Packet Scheduling Algorithm for IPSec Multi-Accelerator Based Systems," in ASAP 2004. Galveston, TX, USA: IEEE Computer Society Press, Sept. 2004, pp. 387-397.

A Design Methodology for HW/SW Security Protocols

Alberto Ferrante

Le telecomunicazioni sono di fondamentale importanza per ogni attività di individui ed aziende. Queste ultime hanno l'esigenza di collegare tra loro sedi anche molto lontane, così come hanno bisogno di comunicare con clienti e partner industriali; gli individui hanno l'esigenza di comunicare tra di loro, con le aziende e con le istituzioni.

La sicurezza e la privacy sono altre esigenze importanti del mondo moderno. La crescente competizione tra le aziende, infatti, richiede un sempre maggior livello di protezione dei dati riservati in loro possesso. Le informazioni riservate degli individui richiedono anch'esse di essere protette dato che le crescenti capacità di trattamento delle stesse le espongono sempre a maggiori rischi.

I requisiti di protezione delle informazioni si rivelano spesso in contrasto con l'esigenza di comunicarle. Di fatto, spedire delle informazioni su un qualsiasi canale di comunicazione, potrebbe renderle accessibili a possibili spie. L'unico modo per risolvere questo contrasto consiste nell'introdurre alcuni meccanismi di protezione dei dati. Ne esistono diversi, per lo più basati sull'utilizzo della crittografia tradizionale, ma negli ultimi anni ne sono stati anche sviluppati alcuni basati sulla crittografia quantistica. Quest'ultima verrà sempre più utilizzata in futuro, sebbene solo per alcune applicazioni, in quanto la crittografia quantistica richiede la presenza di alcune condizioni tecnologiche particolari. La crittografia tradizionale, invece, continuerà probabilmente ad essere utilizzata per le applicazioni più comuni ancora per molti anni. Sia per la crittografia tradizionale, sia per quella quantistica si utilizzano dei protocolli di comunicazione sicura. Questi protocolli integrano gli algoritmi crittografici e permettono di utilizzarli in modo corretto per proteggere le informazioni trasmesse. Di norma, infatti, gli algoritmi crittografici non possono essere utilizzati senza un meccanismo di scambio delle chiavi ed un meccanismo di gestione delle connessioni sicure. Questi sono alcuni dei servizi forniti dai protocolli per la sicurezza.

Siccome i protocolli per la sicurezza sono basati sugli algoritmi crittografici richiedono parecchie risorse computazionali per processare grandi quantità di dati in modo veloce. Per questo si utilizza spesso dello hardware specializzato, al fine di supportare bande passanti di rete elevate. Le normali CPU, infatti, spesso non sono sufficienti a questo scopo. Si potrebbe obiettare che, col passare del tempo, i processori general purpose diverranno sufficientemente potenti per compiere queste operazioni in modo sufficientemente veloce. In realtà, però, la legge di Gilder e quella di Moore prevedono che ciò non avverrà. Secondo la legge di Moore, infatti, le capacità computazionali raddoppiano ogni 18 mesi; secondo la legge di Gilder, invece, la banda di rete disponibile raddoppia ogni 12 mesi.

Lo scopo del lavoro presentato in questa dissertazione è di studiare una metodologia di progettazione di architetture miste hardware/software a supporto dei protocolli per la sicurezza. Durante questo lavoro è stata presa come riferimento la suite di protocolli IPSec (IP sicuro) poiché sta acquistando sempre maggior importanza, data la sua adozione all'interno della nuova versione del protocollo IP, IPv6. IPSec è una suite di protocolli che permette di criptare e/o autenticare i datagrammi IP. Nella parte rimanente di questo testo, vengono presentati i contributi più importanti di questa tesi.

Tramite il profiling di una delle implementazioni di IPSec attualmente utilizzate si sono meglio compresi i requisiti computazionali di questa suite di protocolli. Ciò ha permesso di valutare quando è necessario utilizzare dello hardware specializzato e di derivare delle linee guida per la configurazione di IPSec. E' stato inoltre sviluppato un modello astratto in UML di IPSec utilizzato poi come riferimento durante le successive fasi di design e come punto di partenza per proporre una metodologia di testing per sistemi che implementano IPSec. In questa metodologia, infatti, si utilizzano gli statechart che descrivono gli stati interni dei protocolli per generare dei test pattern tramite dei criteri di copertura dei diagrammi.

E' stato effettuato lo studio di alcune implementazioni attuali di IPSec e per questi sono

state proposte alcune ottimizzazioni: tre differenti algoritmi di scheduling dei pacchetti da processare su core crittografici multipli. Il primo algoritmo è stato pensato per minimizzare il tempo di attesa di ogni pacchetto, ossia il tempo che ognuno di questi deve attendere prima di essere processato. Il secondo consta in una modifica del primo, specificatamente pensata per migliorare il comportamento del sistema durante il processing di pacchetti piccoli; questi ultimi, infatti, causano spesso problemi di performance ai sistemi correntemente utilizzati. Questo è un problema che non può essere trascurato in quanto il 40% dei pacchetti che transitano su Internet è di soli 40 byte. Il terzo algoritmo di scheduling proposto è anch'esso pensato per minimizzare il tempo di attesa dei pacchetti, ma fornisce anche supporto alla Quality of Service.

Le limitazioni precedentemente analizzate, difficilmente superabili con modifiche dei sistemi attuali, hanno portato alla proposta di un innovativo System on Chip (SoC) che permette di processare il traffico IPSec in modo efficiente; di tale SoC è stata sviluppata l'architettura ad alto livello. Questo chip è basato sull'idea di architettura di sistema detta *flow-through*: tutto il traffico, incluso quello non IPSec, passa attraverso il SoC prima di poter essere processato dall'host. In questo modo i pacchetti non richiedono di essere spostati diverse volte tra il processore host e gli acceleratori crittografici e/o di protocollo per essere elaborati. Al fine di minimizzare i trasferimenti dei dati all'interno del chip, si è utilizzato un modello di elaborazione a memoria condivisa. Il limite superiore alla banda supportata dal chip sarà dunque determinato dalla velocità della memoria stessa. La banda passante di quest'ultima dev'essere infatti circa otto volte più grande di quella che si vuole supportare per la rete. Oltre alla memoria condivisa, le principali unità funzionali che sono presenti nel chip sono: due unità di I/O (una per l'interfaccia inbound ed una per l'interfaccia outbound), tre unità di gestione dei database IPSec (una per il database inbound, una per quello outbound ed una per il database utilizzato da IKE), uno scheduler, uno o più core crittografici e un'unità per processare gli header IPSec. IKE può essere sia implementato all'interno del chip, utilizzando un apposito blocco funzionale, sia all'esterno, tramite un'opportuna combinazioni di hardware e di software. Al fine di ottenere performance superiori a quelle ottenibili utilizzando un singolo chip, vengono proposti dei modi per collegare più SoC in parallelo. Viene inoltre mostrata l'architettura interna di due unità funzionali: quella che implementa il protocollo IKE e quella di management dei database di IPSec.

Bibliografia

[1] Antonio Vincenzo Taddeo, Alberto Ferrante, and Vincenzo Piuri, “Scheduling Small Packets in IPSec-based Systems," in CCNC 2006. Las Vegas, NV, USA: IEEE, 8 Jan. 2006.

[2] Alberto Ferrante, Vincenzo Piuri, and Fabien Castanier, “A QoS-enabled Packet Scheduling Algorithm for IPSec Multi-accelerator Based Systems." in Computing Frontiers. Ischia, Italy: ACM, May 2005, pp. 221-229.

[3] Alberto Ferrante, Vincenzo Piuri, and Jeff Owen, “IPSec Hardware Resource Requirements Evaluation," in NGI 2005. Rome, Italy: EuroNGI, 18 Apr. 2005.

[4] Alberto Ferrante, Giuseppe Piscopo, and Stefano Scaldaferri, “Application-driven Optimization of VLIW Architectures: a Hardware-Software Approach," in Real-Time and Embedded Technology Applications. San Francisco, CA, USA: IEEE Computer Society, 7 Mar. 2005, pp. 128-137.

[5] Alberto Ferrante, Uljiana Boiko, Antonietta Lo Duca, and Vincenzo Piuri, “A Testing Methodology for IPSec-based Systems," in SoftCOM 2004. Dubrovnik and Split, Croatia, Venice, Italy: University of Split, Oct. 2004, pp. 22-26, sponsored by the IEEE Communication Society and by the UNESCO Communication and Information Society.

[6] Fabien Castanier, Alberto Ferrante, and Vincenzo Piuri, “A Packet Scheduling Algorithm for IPSec Multi-Accelerator Based Systems," in ASAP 2004. Galveston, TX, USA: IEEE Computer Society Press, Sept. 2004, pp. 387-397.