

Exponentiating Alice¹ and Bob: Building Ultra-High Performance Silicon Security Systems

With the increase in Internet traffic and the subsequent number of users requiring secure transactions, processing bottlenecks occur more frequently. To date, workarounds in hardware, software and services are the proffered solutions. Yet, moving to the next level is inevitable. Transferring the heavy number-crunching from the one-on-one, Alice-and-Bob encrypted transactions to a high-performance security system embedded in silicon requires consideration of certain factors. This paper discusses those considerations.

Ecommerce is expected to grow exponentially. International Data Group's market analysis arm IDC estimates the worldwide ecommerce transactions totaled \$354 billion in 2000. According to the Boston Consulting Group, online buying was less than 2 percent of total retail sales in North America in 2000, yet BCG expects retail revenues from online sales will be \$65 billion in 2001,² making the Web the fastest-growing retail channel.³ As more people come online, especially from outside the U.S., IDC, Forrester Research and BCG estimate the number of online transactions to range between \$5-7 trillion by 2005.⁴

The means of securing electronic transactions exist and have gained wide acceptance. Primarily, Secure Sockets Layer (SSL/TLS) is the de facto standard developed by Netscape Communications for securing transactions over the Internet. SSL/TLS is already built into every browser.

For virtual private networks, the Internet Security Protocol (IPSec) standard prevails. [See **Securing Transactions on Public & Private Networks.**]

Especially for SSL, the larger the encryption key, the more secure the transaction – and the more resources needed to decrypt at the other end of the transaction. Such encryption-based techniques put huge demands on the network and the pipelines. Imagine the resource burden as more and more users come online. No wonder the current SSL techniques are used only for the most sensitive data transfers. [See **Cryptography Specifics.**]

This poses a conundrum. With resource-intensive transactions already taxing systems, how will the Internet tolerate and allow the growth of online transactions, while simultaneously ensuring the security of the data being transferred? How will e-commerce thrive as is projected?

Clearly, there is a need to painlessly process the absolutely huge numbers required for secure online transactions.

Designing the Solution

The technology of scrambling data and messages for electronic communications has become crucial to security for businesses and consumers alike. The need for security is not going away. As it is, the process is fairly intense, requiring enormous efforts to manage.

It used to be that starting a car required muscle and brawn. Cranking it up was a physical act, requiring manipulation external to the car. Now, starting a car is just a matter of a twist of the wrist.

¹ Often, in technical descriptions of encryption processes, “Alice” is the name associated with the encryptor and “Bob” with the decryptor.

² Industry Standard, May 10 2001.

³ The Industry Standard Jun 11 2001.

⁴ IDG, Industry Standard, May 23 2001.

In general, automation of routine processes is essential to growth in any industry. In that regards, securing transactions is no different from starting a car – it must be automated so that ecommerce can grow and thrive.

Ecommerce security is based on running complex mathematics, quickly, efficiently and seamlessly over the Internet, a medium which is accessible to the public and an infinite number of users. Only so many transactions can be processed simultaneously without degradation to the system. And as ecommerce grows, the number of transactions is projected to increase exponentially, thereby imposing even more on the system.

Offloading

One overarching solution to relieve bottlenecks is to offload security processing to other devices. The Web server, which provides the interface between the browser and the application, runs in parallel with the application as a separate task under a real time operating system (RTOS), or as an integral part of the application. The application handles other external hardware and provides the interface to other data acquisition devices much like any embedded system without a Web server. Scalability is implemented by adding servers, cards and load balancing.

Having to monitor through-put at the Web server level and manually expand the network capacity card by card, server by server, while balancing loads is so labor intensive it has spawned a new industry – network management services – and also a huge opportunity for automation.

Then, too, hackers usually break into sites through holes in Web server software. According to a survey of ecommerce sites conducted by the Computer Security Institute, 85 percent experienced a security breach in 2000.⁵

Today's security processors, co-processors and ASICs require application programming interfaces (APIs), RTOS, software and the "glue," such as network processors, host processors, and so on. None stand alone as a panacea.

Given this climate, offloading crucial functions to a Web server, a cryptographic accelerator or cleverly tweaking software to bear the burden is like bailing water out of a sinking boat.

Sharing the burden

Another issue is where transaction processing occurs – on the data plane or the control plane. Traditionally data plane functions are:

- Pattern matching and packet classification
- Packet processing or data modification
- Traffic or queue management and traffic shaping
- Security

Whereas, control plane functions are:

- Set-up/tear-down
- Table updates
- Register/buffer management
- Exception handling

As with many paradigms, functions once housed under one roof grow too complex and are spun off. The same is true for the functions initially placed solely on a single chip, which are now split into others with more specificity. For example, ASICs now take care of either part of all of the datapath; general processors,

⁵ The Industry Standard, June 4, 2001.

the controlpath. A network processor or RISC processor can handle table-look-up functions, for example, but an external memory chip is needed to house all the table information.

When introducing ancillary devices and software where exchanges are made to complete a secured transaction, both planes must be accessed. Obviously this adds to the burdens of the general processor as well as co-processors.

RISC architecture

Network equipment is getting more and more sophisticated as specialized hardware is developed to meet the upsurge in demands made from Internet traffic. For example, servers dedicated to the Web range in value and complexity from a low-end PCI card to a full-fledged, high-end IBM Websphere server.

RISC processors are solving network-processing challenges to date – kind of. But as gigabit rates become the norm, sequential RISC processor bus speeds and clock speeds have not typically kept pace. To help keep up with the new high speeds, specialized hardware are being put online, even though they have limited programmability.

A processor with a bus width of 64 bits that clocks at 50MHz has a peak bus bandwidth of 3.2GBps. Assuming all packet data goes into and out of the processor, and a data rate of 1 Gbs second, means 2GBps of the bus bandwidth is taken up solely by ingress.

Another possibility is to partition by function, such as forwarding and routing/signaling, and delegate to different processors. As more partitioning metamorphoses into parallelism for deploying a single stream of data, maintaining context between processors, and synchronization of tasks become more and more complex.

So software solutions are called in to add support. For example, quality of service queuing (QoS) reorders data for egress priority. Data discard and traffic shaping algorithms are also employed to ease data traffic jams.

RISC, which lacks bandwidth to handle high-speed data without buffering, is not optimized for relieving data buildup. The cache cannot help RISC in communication applications due to constantly changing data. In addition, increasing cache size provides diminishing returns.

Throughput

Throughput is affected by the I/O data streams assigned to the CPU. A general processor handles both as well as other activities. When requests from the server are infrequent and one at a time by one user at a time, a smaller CPU can be used because large throughput and data rates are not an issue.

But step up the requests, and the story is different. For example,

- According to Networkshop, a typical Pentium server configuration running Linux and Apache, which can handle 322 connections per second of standard HTTP traffic at full capacity, fell to about 24 connections per second when handling a full load of SSL traffic. A similar test on a Sun 450 server running Solaris and Apache went from about 500 to 3 connections per second.
- The Meta Group reports that a Web server can only handle 1 to 10 percent of its normal load when processing secure SSL sessions.

Board Level Architectural Considerations

Since the priority is to offload processing, then adding additional hardware and software “fixes” does not offer relief. A co-processor assigned to work with the server and perhaps a network Internet card creates internal traffic for data exchanges among the devices and their various functionalities. In addition, it is extremely difficult to keep time-critical data synchronized and in context. Finally, parallel processing schemes are inherently complex and difficult to scale.

On the other hand, a programmable security network processor that is dedicated to handling the entire security process and administration, offers a multitude of solutions for any given situation where ecommerce is a priority.

Finding a solution at the board level will raise the following issues:

- Memory management. Should architecture scheme use multiple processors accessing common memory? Or should each processor be self-contained with its individual memory?
- How much data should the I/O management port support? Should data be able to flow fast as the pipe? (E.g., the theoretical maximum for a gigabit Ethernet pipe would load with SSL 100,000 transactions per second.)
- If the security engine is as fast as the SSL processing, then how do you classify traffic? Which port goes first: Port 80 (HTTP)? Port 443 (HTTPS)?
- If it's just a security engine, then should a complementary router or a network processor be designed?
- What key length will be standard?

Scalability

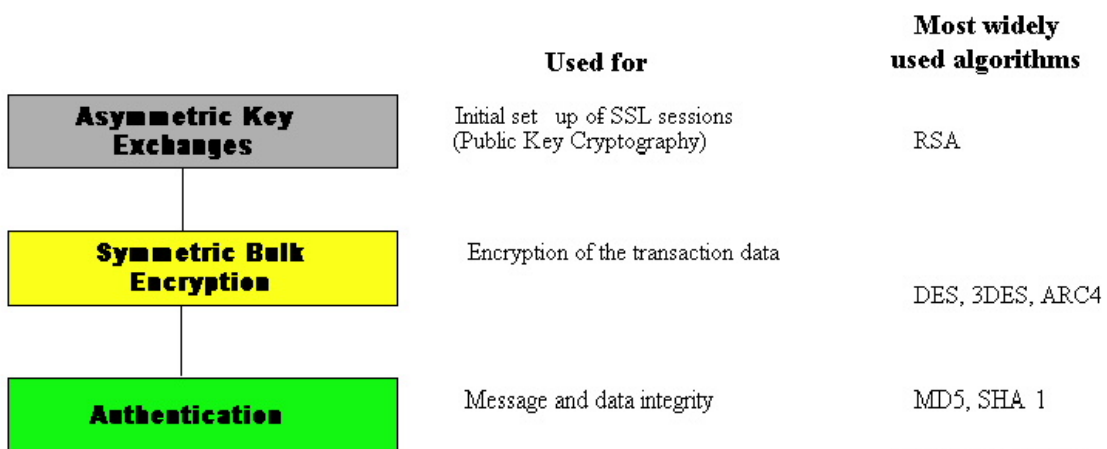
The arms race: As each new generation of processor speed increases, both security designers and hackers have an equivalent game in speed to crack code. Reverse engineering does not work on encrypted material. Instead, standard CPUs are used to hack away at the codes. But by lengthening the key to 1024, 2K or even 4K, the years that it would take to break the key is hardly worth the effort.

Nevertheless, as systems are expanded to handle increased traffic, the limits of scalability are reached quickly when the underlying architecture is not redesigned to explicitly handle much larger data processing requirements.

Doing the Math

SSL encryption-based security is fused from algorithms, such as MD5 and SHA-1 hashing algorithms that provide authentication and data integrity, DES and 3DES algorithms that actually encrypt the data in symmetric bulk encryption, and finally the RSA algorithm that confirm the identity of participants when setting up the secure session using a public key using asymmetric key exchange.

Cryptography Specifics for the SSL Process



Source: Layer N Systems

During the SSL process, the packet bringing the encrypted message is terminated. The message's header is stripped to get to the data and a policy check run. Data is fed in stages to perform hashing and encryption/decryption functions. Then headers are reconstructed. The packet is then ready to service and route.

With the public key algorithm, the encrypting key differs from the decrypting key, which provides additional security. But since everyone is using the same algorithms, key length flexibility becomes a

requirement. A typical key-length is 40 or 128 bits long. The longer the key, for example, a 1024-byte or 2488-byte key is more the norm. Banking and government are now using 4K keys. The longer key lengths means more processing power is required. Needless to say, current chip architectures are inadequate for such sophisticated security mathematics.

Newer Algorithms

Hard at work in private and government labs, mathematicians have developed new schemes for encryption -- elliptic curve encryption; voice encryption; quantum cryptography; and DNA cryptography.

Elliptic curve encryption was developed for use on PDAs and other handheld devices. It uses a smaller key that requires much less memory. Software already exists to provide fully encrypted conversations with any other user connected to the Internet. Quantum cryptography does not introduce a new algorithm, but instead is a means for creating and securing the distribution of private keys. The theory is that that communicating photons cannot be diverted from the intended recipient to the unsought-for interceptor without creating an irreversible change in the quantum states of the system.

In DNA cryptography, human DNA schemes provide the basis for converting a message. A piece of DNA spelling out the message to be encrypted is then synthesized, and the strand is slipped into a normal fragment of human DNA of similar length.

The Security Network Processor

Turning to a silicon-based solution to address the current traffic problem is a natural conclusion. As important, is building the entire networking security system on a chip (SoC) – including all the “mystery and mystique” perpetuated by network security specialists. By extracting the transactions and the Layer 2 and Layer 4 network processing and offloading to a specialized SoC, network administrators can expect extremely high performance that at relatively low maintenance costs.

In addition, system designers do not have to become security experts. Processing is totally self-contained and invisible to the host system. The complexity of security should be hidden from designers who already are taxed to straddle disciplines. The relative scarcity of design engineers makes this point even more important. The design of the processor should be comprehensive enough that the designer does not have to become a security expert or to impose the management of such on another human being like the network administrator.

Designing a durable and enduring solution means it must address the following complementary goals:

- Maximize throughput, while making it easy to implement
- Achieve faster time to market
- Be able to scale with increased demands
- Be OS-independent, while offering best-of-breed options that even security pros approve

Finally, the processor should enable security to be seamlessly and efficiently embedded within the entire Internet infrastructure.

The Bet

Layer N Systems believes that the market requires and will eagerly adopt a silicon security solution that addresses the need for robust security that is optimized for security mathematics, and offers OEMs significant improvements in design architecture, speed, encryption level and throughput

Layer N Systems offers the following:

- Complexity of security is hidden, while offering system architects robust, high-speed solutions.
- Invisible to users too. Millions of users don't have to change their hardware, software or behavior to enjoy higher security levels.

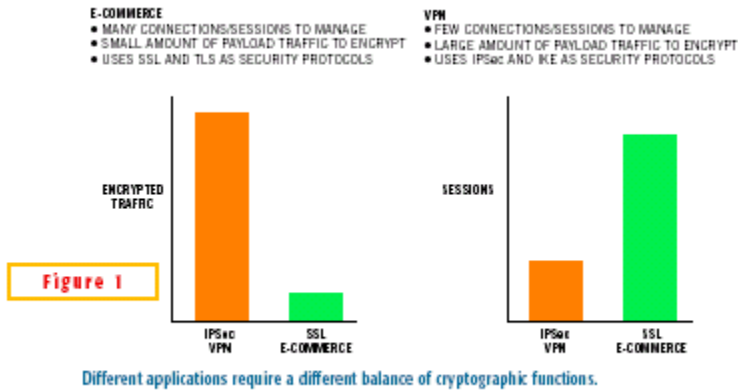
- The future of the Internet as an integral part of the world economy inevitably requires robust, embedded, and transparent security – and that the company that can provide OEMs an intelligent, invisible solution will inevitably be successful.

SIDEBARS

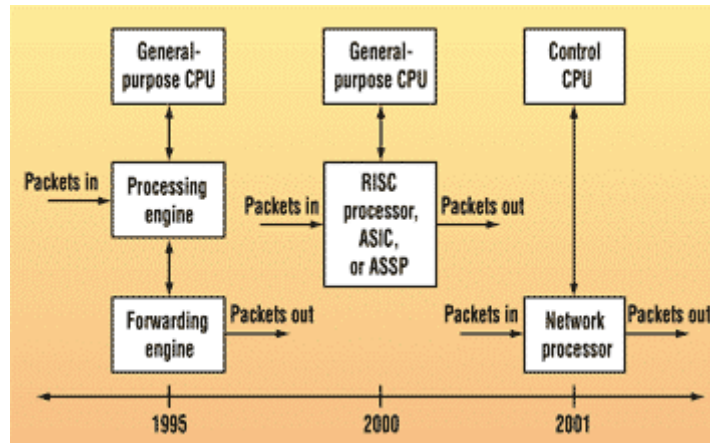
Securing Transactions on Public & Private Networks		
NETWORK TYPE	Public (Internet)	Private (Virtual Private Network)
ENCRYPTION STANDARD	SSL (Secure Sockets Layer)	IPSec (Internet Security Protocol)
DATA SECURED	Select data secured	All data secured within VPN but not between VPNs.
SESSION CHARACTERISTICS	Short-term infinite number of sessions Selected data encrypted	Longer; fewer sessions All data secured within VPN
PROCESS	Create session, send data, tear down session 1. Asymmetric key exchanges 2. Symmetric bulk encryption 3. Authentication [See Figure x. The SSL process.]	VPN to VPN is open. Tunnel between can be secure if within the same administrative network. Otherwise, unprotected.
BOTTLENECKS	-On unsecured servers/accelerators -- bulk encryption; RSA decryption Imbalanced flow – less traffic from client; more to it. On secure servers, setting up and managing the multitude of secure sessions. -Interaction 1. Transaction -- Full setup / resumption / Perpetually active 2. Among devices – external buses, internal (host processor, memory, backplane, fabric) Stateful – more cognizance of each transaction required	Relatively balanced flow-- pervasive bulk encryption with each transaction. Interaction – ongoing within session Stateless – sequence tracking only
FUTURE BOTTLENECKS	New algorithms, stronger keys, and more secure architectures	New algorithms, stronger keys, and more secure architectures

Securing Transactions on Public & Private Networks

CURRENT MAXIMUM THRUPTUT	500 transactions per second → 3 transactions per second	Top end metrics are in the range of 1 Gbps of total throughput, with 500,000 concurrent TCP sessions and 25,000 IPSec tunnels. 35 Mbps. By mid-2002, he says, you will see 10-Gbps chips in production, marking a 300× [is this correct?]
MAXIMUM THRUPTUT WITH SECURITY NETWORK PROCESSOR	100,000 per second for SSL-based.	N/A



⁶ EDN Magazine August, 13, 2001.



The evolution of network system design⁷

A network processor (NP) is a device that is programmable or configurable and optimized to perform networking-specific functions. The term does not apply to ASICs, which have high development costs, time-to-market constraints and short product lifecycles. Nor is an NP an FPGA.

⁷ Apr2001 EDN