

FPGA Implementation of Triple DES

Andrew Watts and Anton Wanio

Abstract

There are many different methods of encryption that have been used throughout history. As encryption methods are cracked, new methods are needed to protect vital data. As these methods become more complex, there is a need to design hardware to take over these CPU intensive tasks. There is a problem however, as encryption standards change, so must the hardware. Utilizing a reprogrammable FPGA will take the CPU intensive encryption tasks off the CPU, and bring them into separate hardware, as well as giving users the ability to upgrade their systems without having to invest in new costly hardware.

Background

A Field Programmable Gate Array (FPGA) is a reprogrammable logic device used as a reprogrammable alternative to ASIC chip devices. FPGAs can be used for specific operational behavior, or general purpose CPU functionality depending on the complexity of the device. FPGA applications include DSP applications, ASIC prototyping, imaging, speech recognition, cryptography, hardware emulation any many other application specific uses. Many FPGAs are implementing shared general purpose CPUs on chip for shorter latency times between operations resulting in higher performance of the overall system. FPGAs also have the capabilities of running loops concurrently thus increasing throughput. Many of the repetitive tasks involved in algorithms such as Triple-DES, AES-128, AES-256, Blow-fish, Two-fish and alike benefit from the capabilities of FPGAs.

There are several encryption algorithms used today. The most widespread algorithm today is the Triple Data Encryption Standard or Triple-DES algorithm. This standard is prevalent in commerce industries, embedded systems applications, non-classified secure defense data as well as secure communication protocols such as SSH,

SFTP, and SHTTP. The Triple-DES algorithm is a revision of the DES algorithm that was created by IBM in 1974 which uses confusion and diffusion to encrypt data. The Triple-DES algorithm is implemented in both hardware and software and is accepted as a widely available standard.

The first step in running a Triple-DES algorithm is key generation. To generate the sixteen sub-keys you must start with a 64-bit seed. This seed is then applied to a permutation table that permutes the seed skipping every 8th bit. This results in a 56-bit primary key. The 56-bit key is broken into upper and lower halves. Each of these halves is run through a loop that, depending on the iteration number, is shifted either by 1-bit or 2-bits creating 16 keys where the nth key is the nth -1 key shifted the appropriate value according to the iteration number. After each half is created the nth upper half and lower halves are concatenated and 16 sub-keys have been created. These keys are permuted a second time against another permutation table that also reduces the size of the keys to 48-bits. These are the final keys needed to encode the data.

To encode the data, the message is broken down into 64-bit blocks. If the block is smaller than 64-bits, it is padded with 0's. A data permutation table is used to perform an initial permutation of the data. The block is then broken into higher and lower halves. At this point the Feistel function is implemented. The Feistel function expands each 32-bit half of the block to 48-bits by duplicating some of the bits in an expansion permutation table that repeats some of the bits. The right half is exclusive ORed with the sub-key and its output is then exclusive ORed with the left half and repeated for 16 iterations. After the right 32-bits of data is exclusive ORed with the key, the resulting pattern is broken up into 8 6-bit segments. Each 6-bit segment is used in a separate corresponding substitution box (S-box). The S-boxes are used to lookup replacement bits. For each six bits that are entered into the S-boxes 4 bits are returned, thus the S-boxes generate a 32 bit output when the results are concatenated. The 32-bit output of the S-boxes is then permuted against a final permutation table resulting in a 32-bit encrypted data block. To decrypt the data the inverse operation is performed. Triple-DES utilizes this same algorithm, potential but applies it 3 times utilizing at least 2 different seeds.

Methods

Utilizing an FPGA and ImpulseC as a development platform there is potential to develop a standard method of implementing a reconfigurable encryption system for embedded and system level applications

Discussion

IPv6 is the replacement Internet Protocol standard to replace the current IPv4 standard. It will be used to increase the number of IP addresses as well as add additional features to incorporate more efficient use of the Internet. IPv6 has encryption notation headers as well as secure authentication capabilities built into the standard putting it at the network layer. This will cause an increase in usage of encryption in commercial commerce. As encryption standards change the algorithms become more complex thus taking more time to execute. Most software implementations utilize extensive CPU cycles to implement these algorithms resulting in loss of productivity for the user of the system. It is imperative that hardware be utilized to implement these types of algorithms for the capabilities of IPv6 to be fully utilized. Utilizing an FPGA for these types of applications is imperative Utilizing an FPGA for these types of applications is imperative for embedded and system level applications so that as the encryption algorithms change so can the hardware without costly replacement of the systems. These devices include credit card machines, ATMs, Cellular Phones, PDAs and other embedded devices, NICs, routers, broadband modems and other devices needing to communicate over secure network and Internet connections.

Expected results

The expectations are to implement the Triple-DES algorithm on an FPGA utilizing ImpulseC as a development environment. This will create a software-hardware

co-developed system that has the potential of being updated as encryption standards change.

Conclusion

Since encryption standards are constantly changing, and new standards such as IPV6 will require encryption and decryption on a regular basis, there is a need for a fast reprogrammable encryption device. This device would be most efficiently implemented using a mix of multi-purpose processors, and FPGAs. The goal of this study is to produce a reprogrammable encryption system utilizing development tools and hardware such as ImpulseC and Xilinx FPGAs.

References

Pasham, Vikram & Trimberger, Steve. (August 03, 2001). "High-Speed DES and Triple DES Encryptor/Decryptor", Xilinx Application Note: Virtex-E Family and Virtex-II Series, XAPP270, v1.0

Field-Programmable Gate Array (7/20/2006), <http://en.wikipedia.org/wiki/FPGA>

Triple DES Encryption (7/20/2006), <http://www.tropsoft.com/strongenc/des3.htm>,

Triple DES (7/20/2006), http://en.wikipedia.org/wiki/Triple_DES

Data Encryption Standard (7/20/2006),
http://en.wikipedia.org/wiki/Data_Encryption_Standard