Reserch On Fast Montgomery's Modular Multiplication Circuits

Toshiaki Yoshihara (110128)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 14, 2003

Keywords: public-key cryptography, RSA, modular multiplication, Montgomery, FPGA.

Recently, encryption technology has become a necessary technology to prevent information on the internet from being monitored by others without authorization.

Secret-key cryptosystems use the same key (the secret key) to encrypt and decrypt a message, but, public-key cryptosystems use one key (the public key) to encrypt a message and a different key (the private key) to decrypt it. In public-key cryptosystems, key delivery and key management is easy, and two or more transmitting persons can also share a common encrypt key. So, public-key cryptosystems being widely researched as a possible ways of encrypting communication for security.

The most important feature of any public-key cryptosystems is ensuring that figuring out the decryption key from the encryption key is as difficult as possible, to prevent unauthorized use of ciphered information. The RSA scheme uses factoring and the elliptic curve encryption scheme uses elliptic curve discrete logarithms. Both of two cryptosystems take a lot of time to determine the key even if we use supercomputer. The safety of cryptosystems are dependent on it.

Most public-key cryptosystems have very large amounts of calculation and they need huge calculation time for encryption decryption rather than the secret-key cryptosystems. In recent days, although speed also with practical software implementation is increasingly obtained by improvement of CPU speed, as at Smart Card the build-in-CPU(8 bits or 16 bits for low ends) processing is slow and is not practical. Software processing cannot expect efficient encryption process, such as an SSL server. Therefore, in a low-end and high-end both, the dedicated hardware for cryptograph is needed.

In this research, we focus attention on the modular operation to a big integer is repeatedly performed by the many public-key cryptosystems. If such an operation part is accelerable, we can think that the whole hardware for public-key cryptosystems can be accelerated. Although many algorithms which performs modular operation are known, in this research we focus on the Montgomery multiplication algorithm. This algorithm is a technique of calculating multiplication with a modulus, without using division processing. Usually, since division takes a lot of processing time mostly in the four-arithmetical-operations operation in multiple length integer, it is one of the most effective algorithm.

Specifically, our hardware outputs $A \cdot B \cdot R^{-1} \bmod N$ where A and B are smaller than N. Where R sets to $2^{r \cdot m-1} \leq N < 2^{r \cdot m} = R$, and for m' multiplications a multiplicand A has

separeted into m'block which base (or radix) is 2^r. It can be used to compute $A \bmod N$ etc. and can be thought to be general-purpose. Since this algorithm is replace division by R with the $r \cdot m$ bit right shift, it has the merit of being easy to implementation in hardware.

In this research, this Montgomery multiplication algorithm is inplemented in FPGA, and improvement in the speed is attained. We use an FPGA with 3 million gates, XC2V3000 of Xilinx which has 18-bit high-speed multiplier was carried was used. The hardware description language Verilog HDL is used to describe the hardware, and the ISE Logic Design Tool of Xilinx is used to compile the souce. By using the tool, the source code written by RTL (level which puts in a clock and is expressing the transmission state (connection state) of a register, a counter and which are the composition element of a circuit, and the data flow between them) is compiled to on the Net-List which is the form which can carry out direct download at FPGA. Moreover, a simulation can actually be performed as hardware and the time concerning actual operation can be measured.

We have developed hardware computing $A \cdot B \cdot R^{-1} \bmod N$ with each of N, A, B having 1024 bits, respectively. Since the FPGA that we used has 18-bit multiplier, we set $r = 16$.

ISE Logic Design Tool was used for the analysis of the hardware. Actual operation time is the time of only calculation and it was taken as the number of clock cycles applied the maximum delay time. Moreover, it compared with the Montgomery multiplication on general software in a different three personal computer environment.

The results show that our hardware attains up to 20 speed-up factor over the software solutions.