THOMAS - A COMPLETE SINGLE CHIP RSA DEVICE

by Dr. Gordon Rankine,
RAANND SYSTEMS Ltd.,
Livingston, EH54 9BJ,
Scotland, Great Britain

Synopsis - This paper examines a novel implementation of a 512-bit modulus exponentiator for applications in RSA key management environments.

The device, known by the internal project code THOMAS, is a complete single chip RSA implementation. No other device is necessary to compute the RSA components, other than the control elements associated with the crypto-system.

The approach chosen is examined to establish the benefits from the implementation in comparison with potentially faster but less flexible techniques.

1. BACKGROUND

In 1985, TALLIS Security, a division of British Telecom, approached RAANND SYSTEMS Ltd. to design, develop and manufacture, a high speed stream encryption device with an RSA key management procedure. Subsequently, the Government and Advanced Projects division of British Telecom adopted the idea and outlined an extension of the work to become a standard Telecom product for medium and high security line communications. This resulted in the product now known as LEKTOR. During the development program, the RSA implementation chosen was a hardware-assisted MC6809. It became apparent during the development that the implementation of the exponentiation could be the basis for a chip to implement a high-speed RSA device. Further development resulted in the design of a device which bore little resemblance to the original idea, being correspondingly faster and more silicon-efficient. Thus - THOMAS was an accident.

2. WHY VLSI RSA?

Before identifying strategies for the successful implementation of single chip modulus exponentiation functions, hereafter to be (erroneously) denoted by RSA, the question as to whether an RSA device is necessary or desirable must be considered.

The RSA algorithm is now well-known, developing from original public key cryptographic methods, first published in [1 - Diffie, Hellman, 1976]. The recognised version of the algorithm conventionally known as RSA is attributed to [2 - Rivest, Shamir, Adleman, 1978]. Subsequent to this, variants using the same mathematical basis have been developed. As these still employ the exponentiation, any device fulfilling the requirements of the RSA public key algorithm automatically satisfies the related applications. Accordingly, the term RSA will also embrace variants that satisfy this criterion.

The strength of an RSA system is based on the factorisation problem associated with the product of large primes. Recent advances, including application of a technique known as the Quadratic Sieve [3 -

A.M. Odlyzko (Ed.): Advances in Cryptology - CRYPTO '86, LNCS 263, pp. 480-487, 1987.

Caron, Silverman, 1986], have shown that factorisation of 80+ digits is now achieveable. These use networks of powerful devices, but still require approximately a month to complete the task, every additional 3 digits approximately doubling the required period. This corresponds to 256 bit (nominal) RSA key pairs. When RSA was first advocated, [2 - Rivest, Shamir, Adleman, 1978], 200 digits were suggested as having the desired security for the forseeable future, corresponding to nominally 600 bits of RSA key pair. Thus, a solution embodying the range of security required, performs 256 to 512 bit operations.

Thus the problem has been established; performing wide, repeated, multiplications and divisions, to achieve a useful operational application of the method. Where software routines have been implemented, except on powerful mini-computers, or hardware-assisted micro-computers, the times achieved for the computation have been poor. Typically, for 512 bit values of average density, an MC6809 requires 4 minutes, an INT8086 70 seconds, and a MC68000 30 seconds. These performances are adequate for key management applications. Many have been implemented, but are less than satisfactory for fast authenticators, rapid key changes using RSA as a transport mechanism, let alone for RSA stream encypherment.

Furthermore, the requirements to perform the operation require the presence of a complete processor sub-system. This may be acceptable in applications where there is a requirement for substantial computing or processing elsewhere, but is an undesirable addition when the remaining requirements are trivial. Not surprisingly, a demand arose for VLSI implementations to achieve orders of magnitude improvement for applications of the nature outlined.

Many papers and much research and development has been devoted to the production of techniques and devices to achieve such performance. These include [4 - Orton, Roy, Scott, Peppard, Tavares, 1986], [5 - Kochanski, 1985], [6 - Rivest, 1985], [7 - Roy, Tavares, Peppard, 1985], [8 - Orton, Peppard, Tavares, 1986], and [9] - Scott, Tavares, Peppard, 1986], and [10 - Beth, Cook, Gollman, 1986]. However, ignoring material not in the public domain associated with hardware implementations of RSA, there are merely a handful of successful implementations, either known to the cryptographic world, or commercially available, albeit embedded in a commercial product. Even these devices, reflect a very recent success, for reasons outlined below. Apart from applications requiring the highest security, the motivation driving VLSI implementations has been cost and performance. The cost savings are reflected in the difference between the components to achieve a desired performance, and a device; the savings in power; and in real estate hence saving in manufacture and test times. Naturally, the savings are offset by the development costs, which in the past have tended to be very substantial.

Notwithstanding the intense academic research associated with cryptography and the establishment of the DES standard, the techniques devised have not achieved the levels of adoption anticipated. This, in turn, has also reduced the interest in the commercial world to develop such devices. However, the advancement of money transfer in many areas and publicity for the success of hackers has rekindled the interest, generating the few RSA implementations that are now available.

With the advent of high performance Digital Signal Processors, e.g. the TMS320 family, a compact, medium performance, device has become generally available for applications of the nature of RSA. NPL [11 - Clayden, 1985] has developed algorithms for the TMS32010 which now execute a 512 bit operation in nominally 2.5 seconds. This has subsequently been enhanced, [12 - Barrett, 1984] and [13 - Barrett, 1986], to typically 1.5 seconds. With further DSP devices appearing on the market, with progressively higher performance, this might appear to detract from the need for VLSI purpose-designed devices. However, whilst such devices give high performance results, these devices consume substantial power, and require additional circuitry and devices to implement the operation. Furthermore, whilst intrinsically flexible by the nature of the algorithm implementation, the device physically is inflexible, and may not be implemented in differing forms to suit particular applications.

Summing up, the climate, the technology, and the need for RSA VLSI devices now coincide!

3. MATHEMATICAL FUNCTIONALITY

The device is required to execute the two functions:-

$$A = B * C \bmod N$$

and

$$A = B ** C \bmod N$$

The conventional usage of the exponentiator is associated with RSA key management operations. However, the device also acts as a high-speed multiplier, with or without modulus correction, for general processing requirements.

4. HISTORIC FAILURE

There have been many attempts to produce devices that perform a high-performance exponentiation function. There have been almost as many failures. These failures may be attributed to the following reasons:-

Exceeding available technology,
Exotic implementation mechanisms,
Ambitious requirements.

Each of these reasons tends to overlap certain areas of the other.

Despite the rapid advances in semiconductor technology, only recently have VLSI chips of 100,000 transistors plus become readily available, particularly to commercial organisations, where yield and cost have been fundamental to the application. Accordingly, the technology for useful implementations has only become available within the space of the last two years. (By useful, the effective bit width of such a device or concatenation of devices is presumed to be substantially greater than 256, typically 512, as above.)

Secondly, the repercussions of the necessity of large bit widths produces a desire to find techniques that overcome the square- or cube-law deterioration in performance as the bit widths increase. These techniques inevitably demand greater areas of silicon, increased power, and poorer yields.

Finally, the poor performance of the software solutions with the need for high-speed solutions has tended to project higher speed requirements on the device. Thus, where key management functions have been the goal, the need for very fast implementations is generally unnecessary.

5. THE LESSONS OF HISTORY

With hindsight, it is the case that demands were made for performance that outstripped the available capabilities. With the substantial improvements in technology, performance requirements may be achieved by less elegant techniques, permitting a wider flexibility. Thus, considering the three reasons for past failures, the goals may be redefined as follows:-

Assume current technology,
Use old and tried mechanisms,
Limit performance.

With the ability to use more transistors on a silicon die, use the largest number available commensurate with desired unit cost. Under these circumstances, the yield may be ignored, with ammortisation of all losses against the acceptable figure.

The RSA algorithms are well known and the requirements for multiplication and modulus division or reduction are well known. Techniques for the execution of these tasks are available for small numbers of bits or groups that are effective and undemanding. If the resultant device, compromised by such unsophisticated techniques, achieves an adequate performance, accept the limitations.

As an overall strategy, set a lower limit on performance that achieves the desired end.

These decisions are all, of course, self evident. Equally, if these targets had been implemented, there would have been RSA devices of a single chip form available for some time, giving a performance of typically 512 bit full exponentiation with 512 bit data and modulus of 10 seconds, well in advance of the DSP devices. Whilst such performance could not be deemed electric, it has only recently been overtaken by the high-performance Data Signalling Processors (DSP) and with a number of associated components (see above).

These represent a basic set of criteria to produce a device. However, to these may be added a further set of requirements that will be shown to complement the criteria, producing a technology component that is flexible for many implementations.

6. IMPLEMENTATION FUNCTIONALITY

The device, known by the internal project code THOMAS, is a complete single chip RSA implementation. "Completeness" was declared to be the complete absence of any other device to perform the RSA computation. Of necessity. there would be other devices to produce a crypto-system of the desired complexity, but not associated with the mathematical operation. In addition, the following criteria were dictated for such a device.

A restriction on the architecture was decreed to ensure an organisation suitable for adequate testing and simulation before committment to silicon, and ATE testing at a wafer level before encapsulation to a high degree of reliability. This architecture was required to exhibit a high degree of flexibility, preferably at the silicon compiler stage, to permit a family of related devices to be produced easily, efficiently, and inexpensively. This results in an implementation that supports any (reasonable) internal bit width without encroaching on the effective bit size of the machine, as perceived by the user or the local processor.

In order to comply with minimal systems, interface characteristics were required to satisfy modern usage, the de facto industry standards.

The packaging options were to be as wide as possible, but offering a choice from very compact (e.g. surface-mount or bonding) to more conventional DIP standards as required.

The design and quality programs were to satisfy both commercial and military standards, if possible.

The implementation be chosen to minimise design time, and risk. This included the simplification of simulation, and the guarantee of a fully operational working device, first time.

7. APPLICATIONS

7.1 KEY MANAGEMENT - LINE SECURITY

RAANND SYSTEMS Ltd., and BRITISH TELECOM, have a series of high performance DES and B-CRYPT stream encryptors for low to high performance line security applications. The use of THOMAS reduces power consumption, real estate, and minimises overheads associated with certification, session key exchanges, and subsequent establishment of further session keys at high frequency. These applications require the 256 to 512 bit effective widths.

7.2 SMARTish CARD

The virtues of authentication and key management apply equally well to areas associated with "SMART cards or intelligent tokens. There are generally restrictions on the number of devices that may be accommodated within a flexible, thin, carrier. However, the security requirements may be even greater, with the portability of the medium. Accordingly, the technology of THOMAS may be used to produce an internal 8-bit or even 4-bit architecture where silicon area is limited, thereby permitting the incorporation of EEPROM, RAM, and processor on the same silicon die.

7.3 STREAM ENCRYPTION

Although typical applications of RSA have not included stream encyphement, such usage offers a highly secure line, even at 512 bit. With the chosen architecture, ready expansion to 1024 bit effective bit width is immediately available. Alternately, the internal bit width may be increased from 64 bits to 128 or other useful values, giving an immediate linear increase in performance.

7.4 CONCATENATION

Where speed requirements or security levels are variable, the same architecture may be used as a slice of the desired word length, but effectively increasing the internal bit width.

8. THOMAS IMPLEMENTATION

THOMAS employs 4 kilobits of RAM / REGISTER storage to hold the following 512 bit (max) parameters:-

Key, Modulus, Data, Result,
and 4 work variables.

The internal structure is based on a 64-bit width, hence the RAM is organised as 64 words of 64 bits.

All ALU functions are based on a 64-bit width operation.

The core of the device employs 5000 standard cells, organised as ALUs, registers, multiplexors, and control logic. This is supported by an integrated RAM array, 64 words x 64 bits. The die is nominally 16mm x 16mm.

All I/O is controlled via 16 control/status and buffer port registers, with automatic internal destination computation performed transparently to the user. As the internal bit width is 64 bits, THOMAS may be configured via the control register to function in multiples of 64 bit slices. Where the data lengths are high, e.g. 512 bits, but the key is small, e.g. 2 to 8 bits, a key length register may be loaded with the significant length of the key to over-ride the default execution time, the algorithm and hence the implementation being wholly symmetric.

Consistent with the desire to minimise external circuitry, an on-board oscillator produces the nominal 20 MHz clock, though external crystal control is permitted. This, in turn, is used to generate lower frequency clocks to drive associated circuitry.

Although the RSA implementation itself has no need for a random number generator, two on-board generators are provided, white noise and a pseudo-random shift register generator, which provides a random output for other uses in a system, via a status register and at package pin.

The operation restores all parameters to the initial state, thereby permitting further data to execute with the same modulus and key, or new data with the same modulus and a newly loaded key.

All I/O is performed on a byte or word-wide basis, user selected or pin configured, with pin configuration of READ / WRITE and ENABLE operation to suit INTEL or MOTOROLA buses.

The implementation produces a cubic relationship for encryption / decryption times. Thus, for a full 512 bit exponentiation with a 512 bit key, the device typically produces the result in 750 milliseconds seconds, whereas a 256 bit x 256 takes 98 milliseconds.

9. THOMAS APPLICATIONS

The device has immediate applications in products that employ RSAkey management. It offers substantial savings in power consumption, volume, and costs. If this were its only application area, this would be substantial. However, the value of the device is the demonstration of a powerful solution, which, by virtue of its internal architcture, lends itself to a series of wider applications.

9.1 HIGH-SPEED RSA

Although THOMAS is an ASIC, using standard celss, simply by increasing expenditure and development time, a full-custom device could have been fabricated. This remains as an opportunity for either wider internal architectures for higher performance, or with the same basic implementation, a smaller silicon area and reduced power dissipation, with a nominal performance improvement.

9.2 INTEGRATION

The architecture chosen is based on an internal 64-bit wide path. Any linear multiple, based on powers of 2, offers scope to reduce the performance and hence decrease throughput linearily. Accordingly, an 8-bit wide pathway increases the execution time by 64/8. Simultaneously, this reduces the core and control logic requirements by approximately 1/6, thereby permitting the introduction of additional components, e.g. EEPROM, ROM, RAM, and a small processor. This offers a high performance single chip key management system and encryptor, ideally suited for SMART-type card applications, battery operated and/or hand-helded devices, or similar applications.

10. CONCLUSION

THOMAS is the first of a family of devices that embody an RSA exponentiation facility for a wide range of applications. The availability of this feature permits the ready incorporation of secure key management in all areas of privacy and high security, with performance as required.

REFERENCES

[1] - W. Diffie, M. Hellman, "New Directions in Cryptography", I.E.E.E. Trans. Information Theory, vol. IT-22, pp. 644-654, November, 1976.

[2] - R. L. Rivest, A. Shamir, and L. Adleman, "On Digital Signatures and Public Key Cryptosystems", Communications ACM, vol. 21, pp120-126, February, 1978.

[3] - T. Caron, R. Silverman, "Parallel Implementation of the Quadratic Sieve", presented at CRYPTO 86, Santa Barbara, CA., August, 1986.

[4] - G. A. Orton, M. P. Roy, P. A. Scott, L. E. Peppard and S. E. Tavares, "VLSI Implementation of Public Key Encryption Algorithms", presented at CRYPTO 86, Santa Barbara, CA., August 1986.

[5] - M. Kochanski, "Developing an RSA Chip", presented at CRYPTO 85, Santa Barbara, CA., August 1985.

[6] - R. L. Rivest, "RSA Chips (Past / Present/Future)", Advances in Cryptology, Proc. of EUROCRYPT 84, pp. 159-165, Springer-Verlag, Berlin, 1985.

[7] - M. P. Roy, S. E. Tavares, and L. E. Peppard, "A CMOS Bit-slice Implementation of the RSA Public Key Encryption Algorithm", Proc. 1985 Can. Conf. on VLSI, Toronto, Canada, pp. 52-55, November 1985.

[8] - G. Orton, L. E. Peppard and S. E. Tavares, "A Fast Asynchronous RSA Encryption Chip", I.E.E.E. Custom Integrated Circuits Conference, Rochester, N.Y., May 1986.

[9] - P. A. Scott, S. E. Tavares and L. E. Peppard, "A Fast LSI Multiplier for GF(2**m)", I.E.E.E. Journal on selected Areas in Communications", vol. SAC-4, pp. 62-66, January 1986.

[10] - T. Beth, B. M. Cook, D. Gollman, "Architectures for Exponentiation in GF(2**n)", presented at CRYPTO 86, Santa Barbara, CA., August, 1986.

[11] - D. o. Clayden, "Some Methods of Computing the RSA modular exponential", NPL Technical Memorandum TTCC 20/85, August 1985.

[12] - P. Barrett, "Communications Authentication and Security using public key encryption", M.Sc. Dissertation, Programming Research Group, Oxford University, Sept. 1984.

[13] - P. Barrett, "Implementing the RSA Algorithm on a standard Digital Signal Processor (DSP)", presented at CRYPTO 86, Santa Barbara, CA., August, 1986.