An Enhanced Secured FPGA based DES

Shivangi Vajpayee
Research Scholar
Department of Electronics & Communication Engineering,
NIIST, Bhopal

Braj Bihari Soni
Assistant Professor
Department of Electronics & Communication Engineering,
NIIST, Bhopal

ABSTRACT
In this paper we demonstrate an efficient and compact reconfigurable hardware implementation of the Data Encryption Standard (DES) algorithm. Our design was implemented on FPGA of device VirtexEXCV400e. As a strategy to reduce the associated design critical path, we utilized a parallel structure that allowed us to compute all the eight DES S-boxes simultaneously. The testing of theimplemented design shows that it is possible to generate data in 16 clock cycles when non-pipelined approach is employed. Whenpipelined approach is employed on the other hand, 17 clocksignals are required for the initial phase only, and one clocksignal is sufficient afterwards for each data generation cycle. TheVery High Speed Integrated Circuit Hardware DescriptionLanguage (VHDL) is used to program the design.

Keywords
DES, FPGA, Parallel structure.

1. INTRODUCTION
Cryptography is the fundamental system to secure computerized data information. As of late because of the overwhelming increment in the volume of data information, secure and quick cryptographic calculations were produced to battle security dangers and efforts to establish safety were thought to be vital wherever, advanced information exchanges must be performed. The hoisted assorted qualities seen on security applications represented an extra test following exceptionally secure calculations were not by any means the only necessity but instead elite for a few applications and for others, less space. In that situation, cryptographic architects have investigated acknowledge on programming stages, as well as on excellent equipment or reconfigurable equipment stages also. Actualizing cryptographic calculations on reconfigurable equipmentgives significant advantages over VLSI (substantial scale coordinated circuits) and programming stages since they offer fast like VLSI and high adaptability like programming. VLSI usage are quick yet must be composed the distance from behavioral portrayal to the physical design. They need to take after a costly and tedious creation process. Programming executions offer high adaptability yet they are not sufficiently quick for the applications where time component is fundamental. Then again, reconfigurable gadgets are appealing subsequent to the time and expenses of VLSI outline and manufacture can be lessened. In addition, they offer high potential for reinventing and probing various architectures or a few modifications of the same building design. Among the distinctive cryptographic calculations, the most mainstream illustration in the field of symmetric figures is the Data Encryption Standard (DES) calculation [1, 2], which was produced by IBM in the mid-seventies. The DES calculation is sorted out in redundant rounds made out of a few piece level operations, for example, legitimate operations, stages, substitutions, shift operations, and so on. In spite of the fact that those components are actually suited for productive usage on reconfigurable gadgets FPGAs, DES executions can be found on all stages: programming [1–5], VLSI [6–8] and reconfigurable equipment utilizing FPGA gadgets [9–13, 8, 14]. In this paper we introduce a proficient and minimized DES construction modeling particularly intended for reconfigurable equipment stages. The DES usage exhibited in this paper contrasts from different past works in the accompanying angle: It makes utilization of an eight DES S-Boxes parallel structure, bringing about a noteworthy diminishment of the basic way for encryption/decoding. Whatever is left of this paper is sorted out as takes after: Section 2 portrays the DES calculation. Our proposed DES structural engineering and its execution on a reconfigurable equipment gadget is exhibited in Section 3. Area 4 contrasts the accomplished results and the past DES usage. Conclusions and future work are attracted Section 5.

2. THE DES ALGORITHM
On August, 1974, IBM presented an applicant (under the name LUCIFER) for cryptographic calculation because of the second call from National Bureau of Standards (NBS), now the National Institute of Standards and Technology (NIST)[15], to secure information amid transmission and capacity. NBS propelled an assessment process with the assistance of National Security Agency (NSA) lastly received on July 1977 an adjustment of LUCIFER calculation as the new Data Encryption Standard (DES). The Data Encryption Standard [16], known as Data Encryption Algorithm (DEA) by the ANSI [17] and the DEA-1 by the ISO [18] remained an overall standard for quite a while and was supplanted by the new Advanced Encryption Standard (AES) on October 2000. In any case, it is normal that DES will stay in people in general area for various years. It gives a premise to correlation for new calculations and it is likewise utilized as a part of IPSec conventions, ATM cell encryption, the protected attachment layer (SSL) convention and in TripleDES. A nitty gritty depiction of the DES calculation can be found at [19–21].

DES is a square figure: It scrambles/unscrambles information in 64-bit pieces utilizing a 64-bit key (in spite of the fact that its powerful key length is as a general rule just 56-bit). DES is a symmetric calculation: The same calculation and key are utilized for both encryption and unscrambling. DES is an iterative figure: the essential building square (a substitution took after by a stage) called a round is rehashed 16 times. For every DES cycle, a sub-key is gotten from the first key utilizing a calculation called key timetable. Key calendar for encryption and unscrambling is the same with the exception of the minor distinction in the request (opposite) of the sub-keys for decoding. An essential calculation stream for scrambling/unscrambling one square of information is appeared in Fig. 1. Encryption starts with an Initial Permutation.

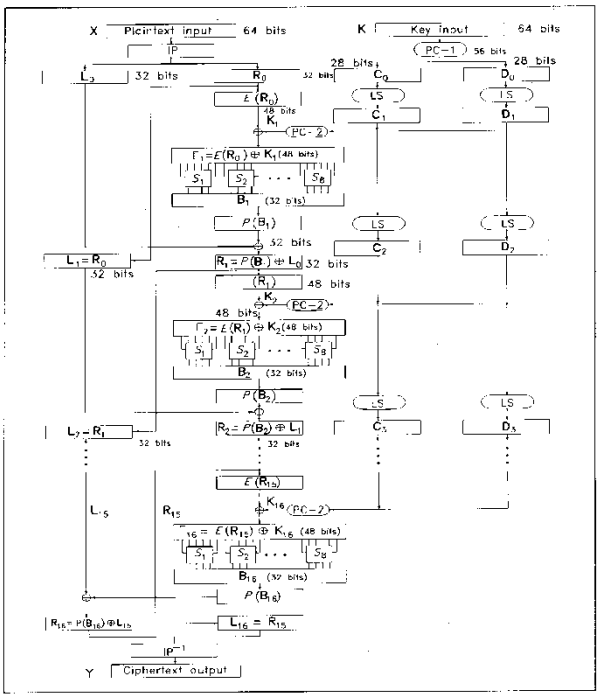

Fig. 3.1 Block cipher design used by DES

Fig. 1.DES Algorithm

which scrambles the 64-bit plain-message in an altered example. The starting's aftereffect change is sent to two 32-bit registers, called the right half enlist and left half enroll. Those registers hold the two parts of the moderate results through progressive 16 cycles. The right's substance half enlist are permuted (change E) and sent to a restrictive OR unit alongside the sub-key for every cycle. Note that a few bits are chosen twice, permitting the 32-bit register to grow to 48 bits. The 48-bit yield of the selective OR square is isolated into eight gatherings (6-bits each) to address eight substitution recollections (S-boxes). A stage P is connected to 32-bit yield from S-boxes and after that sustains into a selective OR square alongside the left's substance half enroll. The yield of this square is built into a makeshift register, finishing up the first emphasis. At the following clock cycle, the impermanent substance registers are built into the right half enroll and past substance of the right half enlist are built into left half enlist. This procedure is rehashed through the entire 16 DES emphases. After 16 emphases, the right half and left half enlist substance are subjected to a last change IP−1, which is the backwards of the starting stage. The yield of IP−1 is the 64-bit figure content.

3. A RECONFIGURABLE HARDWARE DES IMPLEMENTATION

According to Fig. 1, the 16 cycles of the indistinguishable operations are rehashed which go under the name of a capacity f (R,K). DES consolidates first stage, capacity f (R,K), second change, and key calendar for one encryption as appeared in Fig. 2. As it was specified some time recently, DES encryption and decoding schedules are the same. Just the sub's request keys is switched in the event of decryption

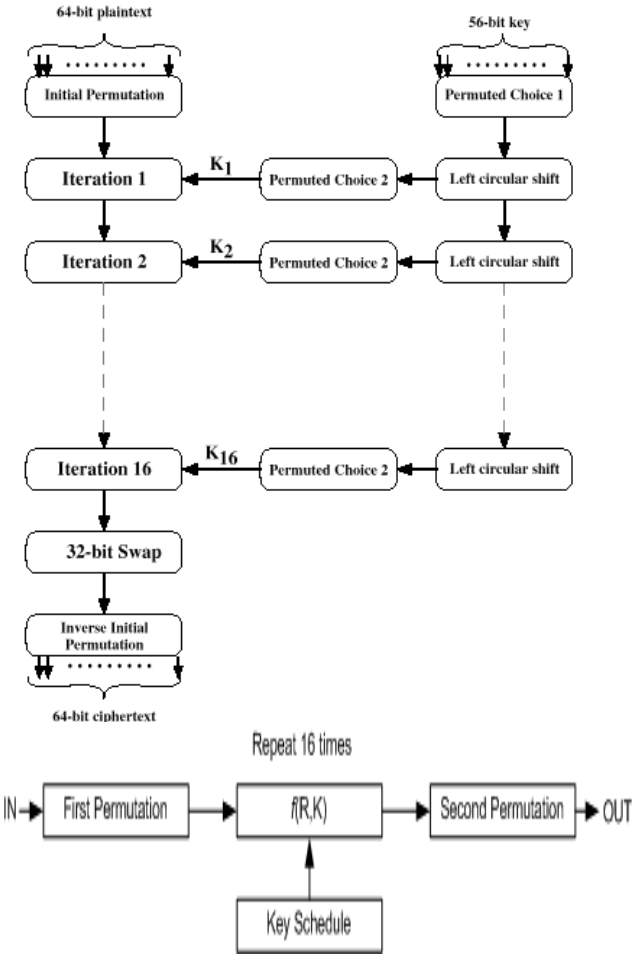

Fig. 2.DES algorithm

Key calendar calculation is straightforward and expends a little time. Also the keys created once, are utilized for the entire session. In any case, in our FPGA execution, pre-processed sub-keys are put away in recollections. In whatever is left of this segment, particular points of interest of the engineering game plan for DES prompting its FPGA execution are exhibited.

3.1 Block Architecture of a DES Round

Fig. 3 speaks to a square outline structural engineering for a DES's round calculation. That construction modeling was particularly custom-made for its effective usage on a reconfigurable equipment stage.DES calculation is fundamentally in light of two operations: altered changes and substitution. Both operations can be capably actualized on a FPGA gadget. DES makes utilization of 8 S-Boxes (each of 64 × 4) possessing an aggregate of 2Kbits. This generally little measure of memory can be actualized by utilizing the appropriated memory assets as a part of FPGAs. Altered stages indeed don't involve FPGA assets as they can be executed by simply changing the wires. Those valuable components were misused to get a proficient usage of DES as appeared in Fig 3. Three inputs: Chip Enable (CE), Clock (CLK), information (IN) and the main yield (OUT) are the four pins of DES chip. Chip empower (CE) enacts the timing rationale and in addition whatever remains of the hardware, in its low state (when it is '0'). The outer clock CLK is the expert clock for the entire circuit that is utilized to create all the control signs to synchronize the information stream.

At the point when CE empowers the circuit, The 64-bit at the info are permuted and isolated into two parts RIN and LIN. At the first rising edge of the clock, both parts are being exchanged to the two's yield registers REGA and REGB. The privilege parts (REGA yield) experience various operations: Permutation E; expansion with sub-key; substitution (through S-Boxes); Permutation P and; expansion with the first left half (REGB yield). Before the following clock comes, the old right half (RIGHT) is the register's info REGB and the new left half (LEFT) is the register's data REGA. The sixteen cycles are then executed. After sixteenth clock cycles the two parts RIGHT and LEFT are linked and the subsequent piece experiences the converse change (IP−1) coming about one encryption for a 64-bit data square. Notice that the utilization of an eight DES S-Boxes parallel structure, results in a noteworthy decrease of the basic way for encryption/decryption.

4. PERFORMANCE COMPARISON

	Base paper	This paper
Journal	**IEEE**	**IEEE**
Device Name	**Vertex**	**Vertex**
CLB'SLICES	**8,192**	**3,778**
Frequency	**228.6MHz**	**397.05MHz**
Throughput	**16Gbps**	**23.10Gbps**
Cycles	**11**	**11**

Table 1 shows the performance figures for some representative DES hardware implementations. Notice that the achieved results are competitive with the existing implementations. A VLSI implementation of DES on static 0.6 micron CMOS technology at [8] is the fastest implementation of DES reported in the literature. Using a pipeline approach, the encryption can be performed at the rate of ≥6.7 Gbs. Several FPGA implementations of DES have been reported in the literature

Achieving throughput ranges from 26 to 10752 Mbits/s using different design strategies. A DES implementation at [12] is a free DES cores which uses pipeline approach in ECB mode and achieves a data rate of 3052 Mbits/s. A java-based (Jbits) DES implementation at [14] achieves the fastest encryption rate of 10752 Mbits/s. DES implementation at [11] implement both 2-stage and 4-stage pipeline approaches obtaining throughput of 183.8 Mbits/s and 402.7 Mbits/s respectively. Almost all FPGA architectures for DES implement use partially or fully pipeline approaches. Only the design in [10] is a one round DES implementation in FPGAs. A fair comparison is possible with this design only. The design was implemented on XC4020E occupying 438 CLB slices. It takes 24 cycles to complete encryption for one single data block achieving a throughput of 26.7 Mbits/s. Hence the Throughput/Area factor is 0.06. Our DES implementation improves both the area and throughput factors consuming only 165 CLB slices on XCV400 and showing a throughput of 274 Mbits/s. The Throughput/Area factor for our design is 2.34. Comparing our architecture with the design in [10], we

get a speedup improvement of almost 10 times in throughput occupying four times less CLB slices. In fact our design ranks second considering as a figure of merit the Throughput/Area Factor is reallyConvincing.

5. GUI

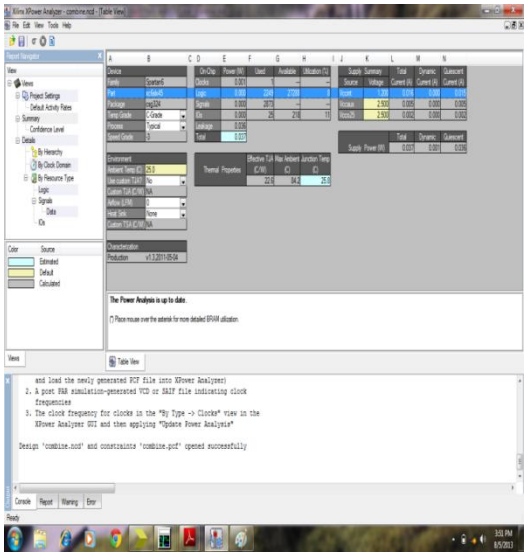

Fig. 4.Combine.ncd and Combine.pcf

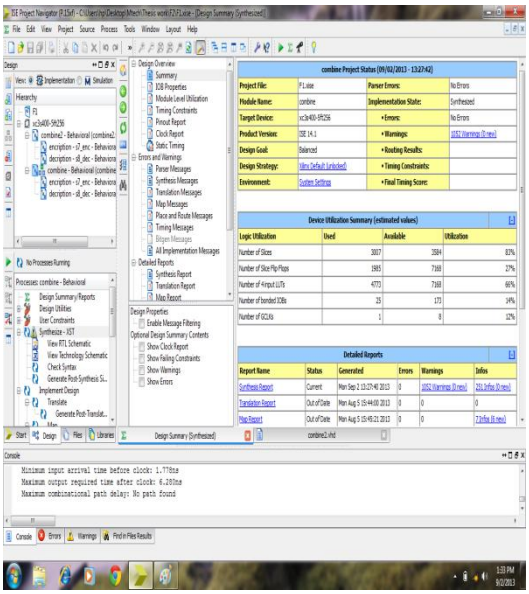

Fig. 5 Design summary (Synthesized)

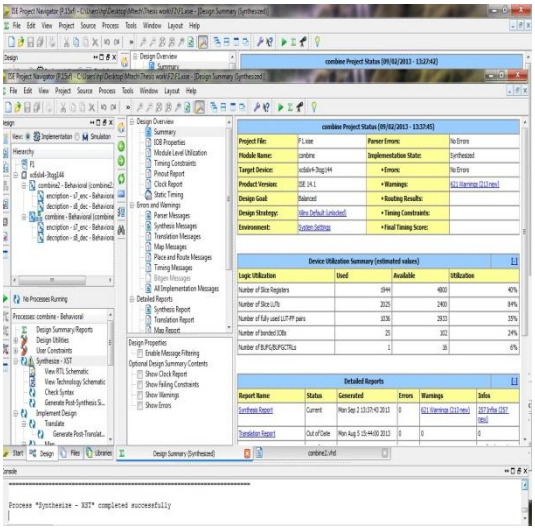

Fig. 7 Design Summary (Synthesized)

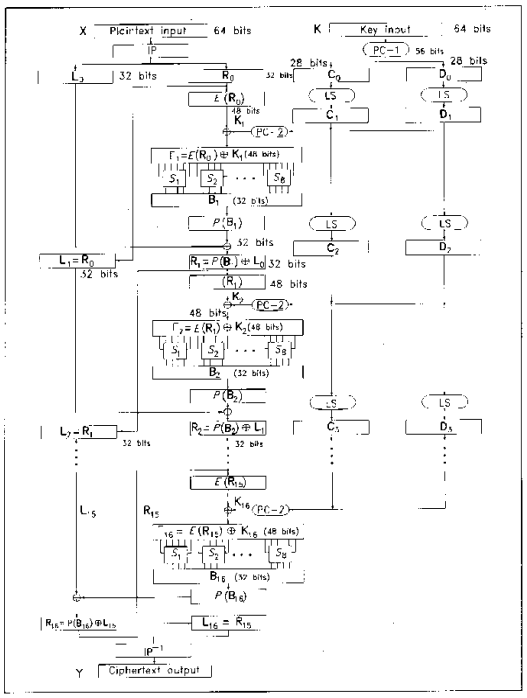

Fig. 3.1 Block cipher design used by DES

Fig. 8 Design Summary (Synthesized)

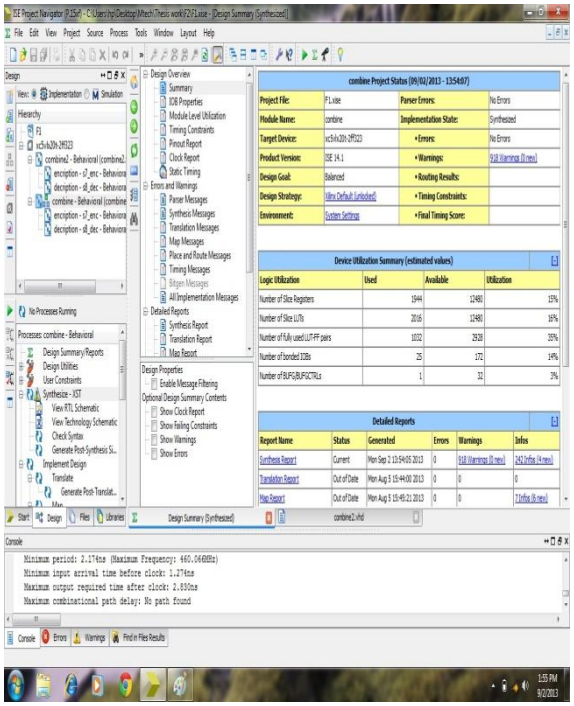

Fig 9 Design Summary (Synthesized)

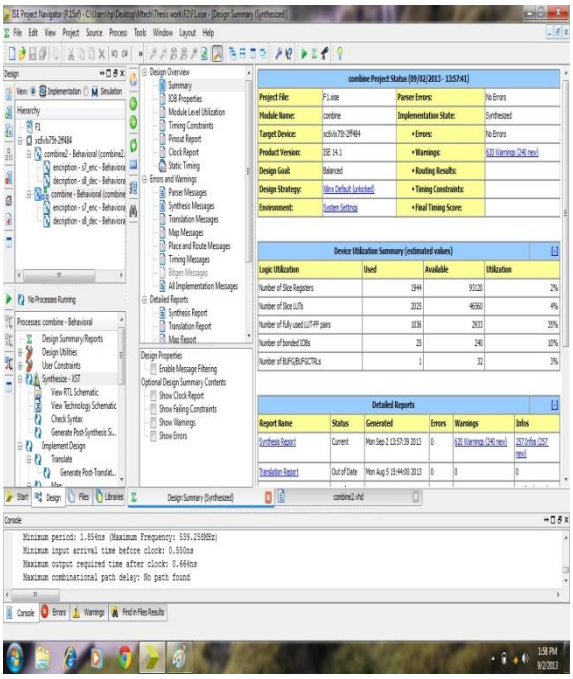

Fig. 10 Design Summary (Synthesized)

All the above GUI shows that the combine DES performance with FPGA, implementation of DES algorithm accomplished on a VirtexE device XCV400e-8-bg560 using Xilinx Foundation Series F4.1i as synthesis tool. The design was coded using VHDL language.

6. CONCLUSIONS

In this work, an efficient and compact DES implementation on reconfigurable hardware platforms was presented. VLSI or FPGA implementations achieve ultra-high throughputs depending on the design strategy; design resources and optimization work both at algorithm and design level. From Table 1, it can be seen that our design achieved a competitive performance when compared with other reported reconfigurable hardware implementations of DES. Our architecture can be improved to offer even better results in terms of achieved throughput. The most obvious extension is to design a fully pipelined architecture in order to obtain a higher throughput at the price of area.

7. REFERENCES

[1] Davio, M., Desmedt, Y., Goubert, J., Hoornaert, F., Quisquater, J.J.: Efficient hardware and software implementations for the DES. In: Proc. of Crypto' 83.(1984) 144–146

[2] Feldmeier, D.C. A high speed crypt program (1989) Technical Memo TM-ARH-013711.

[3] Karn, P.R. (Karns DES implementation source code)

[4] Bishop, M.: An application of a fast data encryption standard implementation.In: Computing Systems, 1(3). (1988) 221–254

[5] Biham, E.: A fast new DES Implementation in Software. In: 4th Int. Workshop on Fast Software Encryption, FSE97, Haifa, Israel, Springer-Verlag, 1997 (1997) 260–271

[6] Eberle, H., Thacker, C.: A 1 Gbit/second GaAs DES chip. In: Proc. IEEE 1992 Custom Integrated Circuits

Conference, New York,USA, Springer-Verlag, 1992 (1992) 19.7/1–4

[7] Eberle, H.: A high speed DES implementation for network applications. In: Advances in Cryptology-CRYPTO'92, Lecture Notes in Computer Science, Berlin, Germany, Springer-Verlag, 1992 (1992) 521–539

[8] Wilcox, D., Pierson, L., Robertson, P., Witzke, E.L., Gass, K.: A DES asic suitable for network encryption at 10 Gbs and beyond. In: CHESS 99, LNCS 1717 (1999) 37–48

[9] Leonard, J., Magione-Smith, W.: A case study of partially evaluated hardware circuits: key specific des. In: Proc. Field-Programmable Logic and Applications,FPL' 97, London, UK, Springer-Verlag, 1997 (1997) 234–247

[10] Wong, K., Wark, M., Dawson, E.: A Single-Chip FPGA Implementation of the Data Encryption Standard (des) Algorithm. In: IEEE Globecom Communication Conf., Sydney, Australia (1998) 827–832

[11] Kaps, J., Paar, C.: Fast DES implementations for FPGAs and its application to a Universal key-search machine. In: Proc. 5th Annual Workshop on selected areas in cryptography-Sac' 98, Ontario, Canada, Springer-Verlag, 1998 (1998) 234–247

[12] Core(2000), F.D.: (2000) URL: http://www.free-ip.com/DES/.

[13] McLoone, M., McCanny, J.: High-performance FPGA implementation of DES using a novel method for implementing the key schedule. IEE Proc.: Circuits, Devices & Systems **150** (2003) 373–378

[14] Patterson, C.: High Performance DES Encryption in Virtex FPGAs using Jbits.In: Field-programmable custom computing machines, FCCM' 00, Napa Valley, CA, USA, IEEE Comput. Soc., CA, USA, 2000 (2000) 113–121

[15] NIST: Announcing the ADVANCED ENCRYPTION STANDARD(AES). Federal Information Standards Publication (2001)

[16] X9.62, A. Federal Information Processing Standard (FIPS) 46, National Bureau Standards (1977)

[17] (Revised):, A.X. National Standards for financial institution key management (wholesale), American Bankers Association (1986)

[18] 8732:, I.D. Banking-key management (wholesale), Association for Payment Clearing Services (1987)

[19] Schneier, B.: Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons, New York (1996)

[20] Menezes, A., Oorschot, P.V., Vanstone, S.: Handbook of Applied Cryptography. CRC Press, Boca Raton, FL (1997)

[21] Trappe, W., Washington, L.: Introduction to Cryptography with Coding Theory. Prentice Hall, Inc., Upper Saddle River, NJ 07458 (2002)