HARDWARE DE CRIPTOGRAFIA E COMPRESSÃO DE DADOS: IMPACTO DE DESEMPENHO[1]

Murilo dos S. Almeida [2]
Luciano Pereira P. [3]

Resumo

A integração de mecanismos de segurança como criptografia de dados a ambientes computacionais eleva a carga de processamento destes, podendo ocasionar um declínio de suas taxas de desempenho. Nesse contexto, essa pesquisa busca mostrar ser possível proporcionar segurança simultaneamente com níveis de desempenho similares aos de ambientes desprovidos de qualquer mecanismo de segurança. Para isso, pretende-se, por meio do emprego da linguagem VHDL e das ferramentas XILINX e MODELSIM, incorporar recursos de algoritmos criptográficos e de compressão de dados em circuitos programáveis (FPGAs), concebendo um hardware, que, ao ser implementado aos dispositivos que compõem um ambiente computacional, proporcionem desempenho e segurança aos dados que por ele trafegam.

Palavras-chave: Segurança, Desempenho, Hardware, Criptografia, Compressão de Dados

Abstract

The integration of security mechanisms as data encryption in computing environments increases the processing load of these, which may cause a decline in their rates of performance. In this context, this research seeks to demonstrate the feasibility providing security both levels and performance similar to environments devoid of security mechanism. To do this, it is intended, through the use of VHDL language and the tools MODELSIM and XILINX, incorporate features of cryptographic algorithms and data compression in programmable circuits (FPGAs), designing a hardware, which, when introduced to devices make up a computing environment, providing performance and security to data travel which it.

Key-words: Security, Performance, Hardware, Cryptographic, Data Compression

[1] Trabalho orientado pelos professores José Eduardo Santarém Segundo e Fábio Dacêncio Pereira dos Cursos de Ciência da Computação e Sistemas de Informação do UNIVEM.
[2] Tecnólogo em Redes de Computadores pelo UNIVEM. Funcionário Público.
[3] Tecnólogo em Redes de Computadores pelo UNIVEM. Técnico em Informática.

INTRODUÇÃO

Atualmente é imprescindível a segurança das informações que trafegam por canais, muitas vezes inseguros, como a Internet.

Para prover segurança a ambientes computacionais, diversas medidas podem ser adotadas, como o emprego de métodos criptográficos, além de outros mecanismos de segurança existentes. Contudo, a utilização de mecanismos de segurança acresce uma carga adicional de processamento a um ambiente computacional, podendo ocasionar queda de seus níveis de desempenho e em suas taxas de transmissão.

Para suprir tal deficiência, propõe-se, por meio de pesquisas, o desenvolvimento de um hardware dedicado a execução de algoritmos criptográficos e de compressão de dados, com o intuito de proporcionar segurança e desempenho a ambientes computacionais.

I METODOLOGIA

O projeto foi dividido em fases descritas na seqüência:

Estudo inicial de todas as tecnologias envolvidas (algoritmos criptográficos e de compressão de dados, sistemas digitais e linguagem VHDL);

Posteriormente o emprego das ferramentas XILINX ISE FOUNDATION e MODELSIM para desenvolvimento, simulação e implementação;

Avaliação dos resultados adquiridos em relação a resultados obtidos em trabalhos correlatos.

O levantamento bibliográfico por meio de uma revisão sistemática de artigos relacionados ao conteúdo deste projeto possibilitou a elaboração de uma tabela com mais de 50 implementações do algoritmo AES (*Advanced Encryption Standard*).

Com isso, pode-se relacionar parâmetros como tecnologia, freqüência, *throughput* e área das implementações. Na tabela 1, tem-se algumas implementações que se destacaram pelo desempenho e consumo de área:

Tabela 1 – Comparação de implementações do AES em FPGAs

Implementações AES			
FPGA	**Freqüência**	**Throughput**	**Área**
Xilinx Virtex II XC2V4000	173.73 MHz	11.12 Gbps	2810 slices
Xilinx Virtex II XC2V4000	95.129 MHz	12.18 Gbps	2518 slices
Xilinx Virtex II XC2V4000	178.17 MHz	22.93 Gbps	3766 slices
Xilinx Virtex II XC2V4000	183.58 MHz	23.50 Gbps	4901 slices
Xilinx Virtex II XC2V4000	184.16 MHz	23.57 Gbps	16938 slices

Fonte: Zambreno (2004)

Na tabela 1, constam diferentes implementações do AES propostas pelo pesquisador Joseph Zambreno, do *Department of Electrical and Computer Engineering Northwestern University* em sua pesquisa realizada no ano de 2004, sobre o desempenho, a área e *delay* do algoritmo AES em FPGAs, de acordo com os resultados mais expressivos divulgados pela pesquisa em questão, em que os modelos de FPGAs adotados tornaram-se muito importantes para comparação dos resultados disponibilizados, em sua grande maioria, relacionados à questão do desempenho.

Por meio da pesquisa de diversos trabalhos vinculados a algoritmos de compressão de dados, foram obtidos diversos tipos de algoritmos para posterior análise, em que as características pertencentes aos mesmos tornaram-se parâmetro de avaliação para escolha dos algoritmos mais apropriados às necessidades deste projeto (eficácia em sua estrutura de funcionamento conjuntamente a índices de compressão satisfatórios).

Após um longo processo de análise, os algoritmos de Huffman e Shannon-Fano foram considerados os mais apropriados para a proposta deste projeto pois possuem uma lógica similar de funcionamento, em que os mesmos realizam o processo de compressão de dados baseando-se na concepção de dicionários adaptativos.

Ao realizar uma minuciosa análise na estrutura funcional dos dois algoritmos anteriormente citados, constatou-se que, embora ambos proporcionassem índices de compressão similares, a lógica aplicada para a geração do dicionário adaptativo pertencente ao algoritmo de Shannon-Fano mostrou-se ser mais eficiente em rela-

ção à lógica pertencente ao algoritmo de Huffman (baseado no emprego de árvores binárias), assim, optou-se pelo emprego do mesmo para realização deste projeto.

2 OBJETIVOS

Fundamenta-se que o aumento excedente da carga de processamento de um ambiente computacional, ocasionado pela inserção de recursos de segurança disponíveis, pode causar decréscimo nas taxas de desempenho e de transmissão de dados pertencentes a este.

Com o objetivo de mostrar ser possível prover índices de segurança satisfatórios conjuntamente a níveis de desempenho similares ao de um ambiente computacional desprovido de recursos de segurança, busca-se, por meio do emprego de circuitos programáveis, combinar as características de métodos de compressão de dados e de métodos criptográficos, para minimizar o impacto no desempenho ocasionado pela inserção de mecanismos de segurança em ambientes computacionais, em que o uso da compressão de dados torna-se responsável por melhorar as taxas de transmissão (GEELNARD, 2006; FONSECA, 2005) pertencentes aos elementos que compõem esse ambiente.

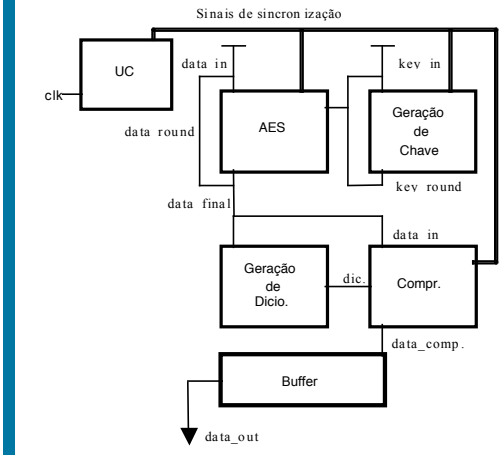
3 ARQUITETURA

A arquitetura é constituída basicamente de um módulo responsável pela encriptação de dados, um módulo dedicado à compressão de dados e um módulo de controle, responsável por gerenciar e sincronizar o funcionamento, além da transmissão de informações entre os demais módulos e dispositivos que compõem um ambiente computacional, como pode ser observado na figura 1.

Para a implementação do módulo criptográfico, optou-se pelo algoritmo de criptografia AES (*Advanced Encryption Standard*), que foi selecionado pelo padrão de criptografia simétrica[3] [4][5]utilizado, além de suas características de funcionamento apropriadas para a implementação em software, bem como em hardware.

Em relação à realização do processo criptográfico, foi empregado para este projeto o uso de blocos de dados de 128 bits.

Figura 1 – Arquitetura básica do módulo de criptografia e compressão de dados



Como anteriormente mencionado, adotou-se para o processo de compressão de dados o algoritmo de Shannon-Fano por suas características gerais de funcionamento (taxas de compressão e a forma de geração de dicionário adaptativo para o processo de compressão), que atendem às exigências relacionadas à proposta do projeto. A exemplo do algoritmo de criptografia AES, empregou-se inicialmente o uso de blocos de dados de 128 bits para o processo de compressão de dados, em que as seções de 4.1 a 4.3, apresentam uma descrição mais detalhada dos principais módulos que constituem a arquitetura proposta.

3.1 AES

O módulo AES é responsável pela realização da criptografia em blocos de 128 bits. Para o processo de criptografia optou-se pelo desenvolvimento de dois componentes principais que são coordenados pela UC; o primeiro componente é o responsável por executar todas as fases correspondentes à criptografia do algoritmo AES: *SubBytes, ShiftRows, MixColumns, AddRoundKey*.

Paralelamente ao seu funcionamento, desenvolveu-se o segundo componente, responsável pela geração de chaves de 128 bits que serão utilizadas durante o processo criptográfico; este é o responsável por

desempenhar as funções de *SubWord*, *RotWord* e *Rcon*.

A UC estabelece o sincronismo entre o funcionamento dos componentes e realiza o controle do fluxo dos valores de entrada e de saída durante a execução do processo criptográfico. A criptografia dos valores de entrada é realizada em *rounds*, sendo necessários dez *rounds* para efetivar o processo. Para cada *round* é utilizada uma chave específica, sendo que essas chaves são obtidas por meio da chave inicial adotada para o processo criptográfico.

3.1.1 Geração de chaves

Para realizar a geração das chaves que serão utilizadas em cada rodada da criptografia de dados, optou-se pela execução contínua deste processo, em que o valor inicial (chave de entrada) é armazenado em uma memória de 128 bits; esse valor armazenado será requisitado sempre que se inicia a criptografia de um novo bloco de dados.

Com exceção do primeiro *round*, toda vez que é iniciada a geração de uma nova chave, utiliza-se a chave do *round* anterior para obtenção da chave da rodada atual. Para realizar esse processo, foi desenvolvido um sistema de realimentação do componente, em que o número de rounds é utilizado como parâmetro de controle para realizar a leitura do valor armazenado em memória.

3.1.2 Criptografia

O componente responsável por realizar a criptografia dos blocos de dados de entrada foi desenvolvido com etapas de funcionamento bem definidas, o mesmo é constituído de componentes específicos para cada etapa do processo criptográfico do algoritmo AES.

As três primeiras etapas do processo criptográfico (*SubBytes, ShiftRows, MixColumns*), são realizadas somente com o valor referente ao bloco de entrada. Na quarta etapa (*AddRoundKey*), torna-se necessária a utilização da chave pertencente à rodada em execução, propiciada pelo componente de geração de chaves. Para que a criptografia de dados ocorra de forma plena, prima-se pelo correto sincronismo entre ambos os componentes.

Antes que se inicie o primeiro *round*, realiza-se a operação XOR (*AddRoundKey*) entre o valor do bloco de entrada e da chave criptográfica utilizada. Ao final do processo criptográfico, durante a execução do último *round*, não é realizada a terceira etapa (*MixColumns*), para que posteriormente possa ser realizado o processo de decriptação de dados. A realização das funções anteriormente citadas torna-se possível por meio de referência ao número de *rounds*.

Assim como o componente de geração de chaves, o componente de criptografia possui um sistema de realimentação similar, baseado no valor correspondente ao número de *rounds*.

3.2 Shannon-Fano

Esse módulo de compressão de dados é constituído de um componente para geração do dicionário adaptativo, um componente de compressão que utiliza o dicionário adaptativo gerado para efetuar a compressão e um buffer para armazenamento de informação processada, em que os mesmos são coordenados pela UC.

Para realizar o processo de compressão, foi estabelecido o uso de blocos de dados de 128 bits e, para o componente responsável por gerar o dicionário adaptativo, empregou-se o byte como unidade de averiguação de redundâncias. Após terem sido gerados, os valores contidos no dicionário adaptativo são utilizados pelo componente de compressão para realizar o processo de compressão; posteriormente o resultado obtido é armazenado em um *buffer* até que seja adquirida uma quantidade de dados compatível ao bloco de dados de saída (128 bits).

3.2.1 Dicionário adaptativo

Este componente avalia o número de redundâncias existentes em blocos de dados de 128 bits, utilizando como unidade de comparação o byte. No processo inicial pela busca de redundâncias, emprega-se o

primeiro byte como unidade de comparação. Esse processo consiste na execução de uma operação *xor* entre o byte de comparação e os demais bytes que compõem o bloco corrente. Posteriormente, executa-se entre os bits resultantes do processo anteriormente descrito uma série de operações *or* que resultará em um único bit ('0' corresponde a um valor redundante, '1' corresponde a um valor diferente).

A cada nova redundância encontrada no processo de averiguação, é incrementado em '1' ao valor referente a essa redundância; e esse processo ocorre até que todos os bytes que constituem o bloco sejam avaliados. Após concluída a etapa anteriormente descrita, as redundâncias são organizadas em ordem decrescente, utilizando como parâmetro de ordenação a quantidade de ocorrências referenciadas as mesmas. Por meio desse processo de ordenação, torna-se possível associar aos valores de maior ocorrência valores constituídos por uma cadeia menor de bits (código de Shannon-Fano), pertencentes a um dicionário adaptativo de prefixo. Posteriormente, essas informações serão disponibilizadas ao módulo de compressão de dados para a efetivação do processo de compressão.

3.2.2 Compressão de dados

Este módulo tem seu funcionamento dividido em duas etapas. A primeira etapa realiza-se por meio de constantes consultas ao dicionário adaptativo, em que se executa a busca de valores pertencentes ao bloco de dados e seus respectivos valores associados durante o processo de geração de dicionário adaptativo (código de Shannon-Fano).

A segunda etapa consiste na substituição de cada byte existente em um bloco de dados pelo seu respectivo valor associado, conforme as informações existentes no dicionário adaptativo. Após um bloco ser totalmente processado, é caracterizado o processo de compressão de dados.

Ao ser concretizada a compressão de um bloco de dados, torna-se necessário o armazenamento do mesmo juntamente com o seu respectivo dicionário adaptativo, para que, posteriormente, seja realizado o processo de descompressão.

3.3 Unidade de Controle (UC)

A UC realiza o sincronismo entre os módulos, além de gerenciar a entrada e saída de dados por meio de sinais de controle. Esses sinais são responsáveis pela habilitação para entrada de um bloco de dados e por disponibilizar o mesmo para a saída, após terem sido concluídos o processo criptográfico e de compressão de dados.

O sincronismo dos componentes que constituem o módulo criptográfico torna-se possível pela constante atualização do valor corresponde ao número de *rounds*, elemento existente nesse módulo; esse valor é alterado em função do sinal de *clock* que alimenta a UC.

Em relação ao módulo de compressão de dados, a UC realiza o gerenciamento da entrada e saída de dados, e coordena o armazenamento de informações no buffer (dicionário adaptativo, tamanho do bloco comprimido e bloco comprimido) de maneira organizada.

Quando o buffer encontra-se com um nível de informações correspondente ao valor estipulado para saída do módulo, a UC, por meio de sinalização, disponibiliza esse conteúdo para a porta de saída. Assim como no módulo criptográfico, a realização dessas funções torna-se possível pelo uso do sinal de *clock*, utilizado como unidade de referência para o gerenciamento das atividades desse módulo.

4 ANÁLISE DE DESEMPENHO

O projeto encontra-se em fase de desenvolvimento e os resultados especificados na seqüência referem-se ao desenvolvimento e à implementação do módulo de criptografia.

A tabela 2 apresenta os resultados iniciais da implementação do módulo criptográfico, em que o mesmo encontra-se em fase de testes e sujeito a alterações em sua estrutura lógica para otimização de seu funcionamento, economia de recursos do FPGA e aumento de *throughput.*

Uma outra alteração que o módulo criptográfico pode estar sujeito é em rela-

ção à integração ao módulo de compressão, para que haja um correto fluxo de informação entre ambos.

Tabela 2 – Resultados da implementação do módulo criptográfico

Estatísticas de Recursos Utilizados do FPGA	
Slices	1403
Flip Flops	702
Luts	3815
IOBs	132
BRAMs	1
GCLKs	3
Temporização	
Tempo de Lógica	5.594 ns
Tempo de Roteamento	0.920 ns
Tempo Total	6.514 ns
Freqüência	
228.781MHz	
Throughput	
1.420 Gbits/s	

Os resultados obtidos com a implementação inicial do módulo criptográfico, principalmente os relacionados à freqüência e ao consumo de área, equiparam-se aos resultados que mais se destacaram em trabalhos correlatos, como pode ser observado na tabela 1. Para isso, o módulo criptográfico foi implementado e simulado no mesmo modelo de FPGA (Xilinx Virtex II XC2V4000) empregado nos trabalhos divulgados, em que a implementação no modelo referido foi de fundamental importância para que pudesse ser efetuada comparação dos resultados obtidos em relação aos resultados divulgados nos trabalhos correlatos.

O módulo de compressão de dados encontra-se atualmente em fase de desenvolvimento, em que a lógica funcional do algoritmo de compressão de Shannon-Fano vem sendo desenvolvida para hardware.

Durante o desenvolvimento dos componentes que integram o módulo de compressão de dados, prima-se pelo desempenho dos mesmos, principalmente pelo módulo responsável pela geração do dicionário adaptativo, pois, como não é possível determinar o número correto de redundâncias em um determinado bloco de dados, almeja-se que o mesmo seja capaz de gerar o dicionário em um período de tempo padrão independente das variações de redundância em um determinado bloco de dados.

O módulo de compressão terá seus índices de compressão em relação ao número de exemplares que compõem o dicionário adaptativo, em que, quanto maior o número de exemplares, menor serão os índices de compressão.

Após a conclusão do módulo de compressão de dados, pretende-se realizar as adaptações necessárias para a interconexão dos módulos de criptografia e compressão e realização de um extenso processo de análise para o recolhimento de resultados mais expressivos, como freqüência e *throughput* do mesmo. Como parâmetro para esses testes, será estabelecido como referência uma rede 10/100 Mbits, com o intuito de se obter e analisar os seguintes valores:

índice de vazão de uma rede desprovida de mecanismos de segurança;

índice de vazão de uma rede provida de mecanismos de segurança (criptografia);

índice de vazão de uma rede com o dispositivo proposto implementado (criptografia + compressão).

Embora o módulo de compressão não esteja ainda concluído, elaborou-se uma estimativa de quanto eficiente torna-se necessário o funcionamento do mesmo para que haja um maior desempenho. Os elementos escolhidos para essa estimativa são os seguintes:

VD => Volume de Dados;
VR => Vasão da Rede = 100 Mbits;
TxC => Taxa de Compressão;
TComp => Tempo de Compressão;
TCrip => Tempo de Criptografia.

A variação dos valores referentes à taxa de compressão, ao volume de dados e ao tempo de compressão especificados na tabela 3 permite realizar uma análise dos resultados obtidos, para que, posteriormente, se possa estabelecer uma estimativa de quanto eficiente deve ser processo de compressão e o tempo de funcionamento do mesmo.

Tabela 3 – Valores para simulação do dispositivo

Teste	VD	TxC	TCrip	TComp
1	80 Gbits	30%	1420 Gbits/s	1500 Gbits/s
2	200 Gbits	50%	1420 Gbits/s	1650 Gbits/s
3	570 Gbits	70%	1420 Gbits/s	1800 Gbits/s

Considerando as informações existentes na tabela 3, foram obtidos os seguintes valores para cada teste:

Teste 1
Tempo de Transmissão de Dados Puros = 800 s
Tempo de Transmissão de Dados Cifrados = 856,33 s
Tempo de Transmissão de Dados Comprimidos = 613,33 s
Tempo Transmissão Total = 652,76 s

Teste 2
Tempo de Transmissão de Dados Puros = 2.000 s
Tempo de Transmissão de Dados Cifrados = 2.148,84 s
Tempo de Transmissão de Dados Comprimidos = 1.121,21 s
Tempo Transmissão Total = 1.263,21 s

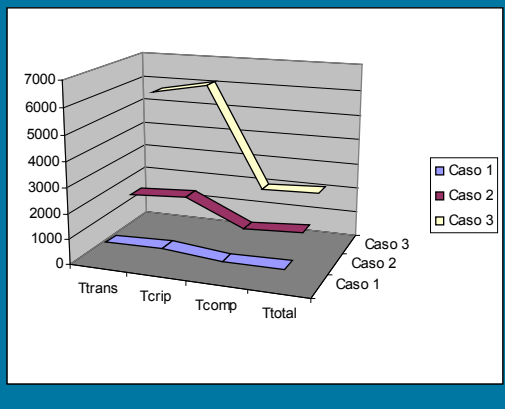
Teste 3
Tempo de Transmissão de Dados Puros = 5.700 s
Tempo de Transmissão de Dados Cifrados = 6.101,4 s
Tempo de Transmissão de Dados Comprimidos = 2.026,66 s
Tempo Transmissão Total = 2.147,08 s

O tempo de transmissão total é o resultado da soma dos valores relacionados ao tempo de compressão, ao tempo de cifragem de dados comprimidos e ao tempo de transmissão de dados comprimidos, em que o resultado obtido proporcionou a concepção de um gráfico contendo todos os dados para posterior análise.

Observando o gráfico 1, nota-se que, embora o processo criptográfico acresça uma carga adicional ao tempo de transmissão, o processo de compressão torna-se capaz de reduzir não só essa carga excedente como também o próprio volume de dados original.

Percebe-se que, mesmo com variações em relação ao tempo de compressão e ao índice de compressão do volume de

Gráfico 1 – Índices de transmissão em rede 10/100 Mbits



dados apresentados na tabela 3, é o valor do índice de compressão obtido que proporciona decréscimo do tempo de criptografia; o mesmo foi atribuído para tabela 3 baseado em índices normalmente obtidos pelo algoritmo de Shannon-Fano. Torna-se importante saber que a variação nos índices de compressão ocorre de acordo com o tamanho do volume de dados a ser processado, em que, quanto maior for o volume de dados, maior será os índices de compressão desse volume, conforme gráfico 1.

Embora o volume de dados para compressão acabe variando entre os testes apresentados, de acordo com a sua proporção, e, embora se tenha atribuído diferentes valores para o tempo de compressão entre os diferentes testes, necessita-se que o resultado final do tempo de compressão obtido seja hábil para que não afete o funcionamento do módulo criptográfico, tornando-o ocioso e elevando, assim, o índice final de transmissão.

CONCLUSÃO

A utilização de circuitos programáveis para o desenvolvimento de aplicações em ambientes computacionais, como sistemas de segurança, pode ser um importante fator para ganho de desempenho desses ambientes, devido à otimização dos mesmos por meio da utilização do paralelismo em *hardware*, em que o emprego de algoritmos de compressão de dados em aplicações de segurança pode proporcio-

nar um ponto de equilíbrio entre segurança e desempenho.

TRABALHOS FUTUROS

O sistema proposto pode ser otimizado com o uso de técnicas de paralelismo como o modelo *pipeline*, em que o uso de circuitos programáveis possibilitaria futuras adaptações no dispositivo em relação ao tipo de aplicação requerida, por meio de utilização de reconfiguração parcial, provendo, além de segurança e desempenho, flexibilidade aos sistemas portadores dessa tecnologia.

O projeto pode ser utilizado em forma de IPs para sistemas embarcados que pretendem utilizar funções de criptografia e compressão de dados combinados, em que novos algoritmos de compressão e criptografia serão desenvolvidos no decorrer do projeto, como: RSA e algoritmos de compressão de imagens e vídeo.

REFERÊNCIAS

GEELNARD, M. **Basic compression library,** [S.l.: S.n., 2006]. Disponível em: <http://bcl.comli.eu/support-en.html> Acesso em: 17 novembro 2008.

FONSECA, T.L.A. **Implementação de um algoritmo de compressão em hardware**. 2005. 62 f. Trabalho de Conclusão de Curso (Engenharia da Computação) - Escola Politécnica de Pernambuco, 2005.

AZIZ, A., IKRAM, N. **Hardware implementation of AES-CCM for robust secure wireless Network**. In: **Annual ISSA Information Security Conference,** *ISSA, 5., 2005*, Johannesburg, Anais: Johannesburg, 2005. p. 44-51

ZAMBRENO, J., NGUYEN, D. A. CHOUDHARY**, Exploring area/delay tradeoffs in an AES FPGA implementation**, In:**International Conference on Field-Programmable Logic**, FPL, 14., 2004, Leuven, Anais: Leuven: Springer, 2004. p. 575-585

MORENO, E. D., PEREIRA, F. D. **Criptografia em software e em hardware**. 1. ed. São Paulo: Novatec, 2005. 288p