SEGURANÇA USANDO CRIPTOGRAFIA ASSIMÉTRICA EM HARDWARE

MUZZI, Fernando Augusto Garcia; TAMAE, Rodrigo Yoshio; ROSA; Adriano Justino

Docentes da Faculdade de Ciências Gerenciais e Jurídicas de Garça – FAEG/Garça

fagmuzzi@yahoo.com.br ; rytamae@yahoo.com.br; adriano@faef.br

RESUMO

Neste projeto implementou-se a especificação do PKCS#11 em hardware, foi utilizado o VHDL e FPGA. Foi implementado por meio de uma máquina de estados finito, usando o algoritmo RSA, formando o projeto modular e facilmente adaptável a expansões futuras para a comunicação entre máquina e dispositivos. Qualquer algoritmo de criptografia pode ser implementado dentro da especificação PKCS#11, e, neste projeto implementou-se o algoritmo RSA. Além da implementação da FSM foi possível realizar implementação e testes usando chave do RSA de 56 bits, 128 bits, 256 bits e 512 bits.

ABSTRACT

In this paper we have designed the PKCS#11 specification on hardware, specifically, for using VHDL and FPGAs. It was implemented through a finite state machines and it works together to RSA algorithm which also was implemented for us into the FPGAs. Our implementation is modular, it is easily adaptable to future expansions for the communication between machine and devices, thus any cryptography algorithm can be implemented inside of the specification PKCS#11. Our results were obtained for the RSA algorithm working with 56, 128, 256 and 512 bits.

1. INTRODUÇÃO

O grande gargalo nas comunicações tem sido os equipamentos de rede, principalmente os roteadores. Eles centralizam o tráfego de pacotes e muitas vezes são os responsáveis pela falta de eficiência da rede. As tecnologias de rede têm

evoluído, principalmente com os avanços da internet, e os roteadores precisam aumentar sua capacidade de processamento dos pacotes, para evoluir juntamente com as demais tecnologias e deixar aos pouco, de ser um dos gargalos das comunicações.

Com a implementação em hardware das especificações do PKCS#11 que propiciará a criptografia, um processador de rede poderá receber e transmitir pacotes criptografados seguindo o padrão.

A principal contribuição deste artigo é apresentar detalhes de nossa implementação em hardware (FPGAs) do padrão PKCS#11. Os resultados usando o algoritmo RSA mostram a viabilidade do projeto e a possibilidade de inserir módulo em outros protótipos em hardware.

2. O PADRÃO PKCS

O PKCS Padrão de criptografia de Chave Pública (*Public Key Cryptography Standards*) é uma série de especificações produzidas pelos Laboratórios RSA em cooperação com desenvolvedores de sistemas de segurança de várias partes do mundo, que visa acelerar, por meio da padronização, a utilização e o desenvolvimento de algoritmos de chave pública (RSA, 2002).

Os PKCS visam preencher o vazio que existe nas normas internacionais relativamente a formatos para transferência

3. O ALGORITMO RSA

O RSA é um sistema de criptografia de chave assimétrica ou criptografia de chave pública que foi inventado por volta de 1977 pelos professores do MIT (*Massachsetts institute of Technology*) Ronald Rivest, Adi Shamir e o professor Leonard Adleman da USC (*University of Southern Califórnia*) (RSA, 2002).

O sistema consiste em gerar uma chave pública (geralmente utilizada para cifrar os dados) e uma chave privada (utilizada para decifrar os dados) através de números primos grandes, o que dificulta a obtenção de uma chave a partir da outra.

O algoritmo RSA usado para a geração da chave pública e privada usadas para cifrar e decifrar as mensagens são simples. (RSA, 2002) (CHIARAMONTE, 2003).

4. ESPECIFICAÇÃO DO PKCS#11

O algoritmo do PKCS#11 da RSA foi implementado em linguagem C (RSA, 2002).

A maioria das chaves utilizadas nos dias de hoje tem 1024 bits de comprimento (RSA, 2002) e quando a implementação é em hardware, usa-se a norma PKCS#11, que é o mais utilizado em interfaces API *(Application Programming Interface*) para módulo criptográfico. Este padrão especifica uma interface (API), chamado Cryptoki, para dispositivos que fazem segurança usando criptografia e executam funções de criptografia usando uma chave que é chamada de crypto-chave (RSA, 2002).

5. O PADRÃO PKCS#11 EM HARDWARE

O padrão PKCS#11 é utilizado para operações criptográficas em hardware. O PKCS#11 é baseado no padrão que fornece recomendações para a execução de criptografia baseada em chave pública e o algoritmo é o RSA. Esta seção apresenta uma versão em hardware do PKCS#11, projetado e implementado pelos autores deste artigo.

5.1 IMPORTÂNCIA DO PKCS#11 EM HARDWARE

O padrão PKCS#11 é utilizado para operações criptográficas em hardware (tokens, smart cards e etc.) para prover suporte aos tokens.

O PKCS#11 é baseado no padrão que fornece recomendações para a execução de criptografia baseada em chave pública e o algoritmo utilizado é o RSA.

Devido a seu pioneirismo e simplicidade do algoritmo, tornaram-se padrão de fato em PKIs.

A segurança da informação se torna mais importante no mundo de hoje, e é necessário que o equipamento de networking permita funções de criptografia.

5.2 DESCRIÇÃO DA MÁQUINA DE ESTADOS FINITO PADRÃO PKCS#11

Nós realizamos o projeto e implementação do padrão PKCS#11 em hardware, especificamente em FPGA. Em nossa implementação, os estados variam do estado 0 até o estado 6, ou seja, a máquina de estado finito RSA padrão PKCS#11 é constituída de 7 estados que estão relacionados entre si. Em cada momento dependendo de um evento, seja baseado no *clock* ou no estado da máquina, um estado estará em funcionamento e passará para o estado seguinte assim que o estado anterior for completado e a respectiva condição for satisfeita.

No estado 5 e 6 pode-se visualizar que é realizado a decifragem, mesmo após a realização de cifragem, isso ocorre para validar os testes, porque foi implementado

para realizar teste em um único FPGA, podendo em projetos futuros usar dois FPGAs, um para cifrar e outro para decifrar.

A figura 1 mostra a simulação usando a máquina de estados finito na ferramenta da Xilinx 3.1 chamada FPGA Express para efetuar simulação. A variável VAR_CH recebe o dado a ser criptografado ou seja 6768 que corresponde aos bytes CD e nota-se que D_OUT no estado 2 tem uma subida de borda indicando que o dado foi enviado serialmente para outro computador. No estado 4 S_D_CRIP recebe o dado cifrado ou seja 1A68FE e D_OUT tem uma subida de borda indicando que o dado cifrado foi enviado pela porta serial para outro computador.

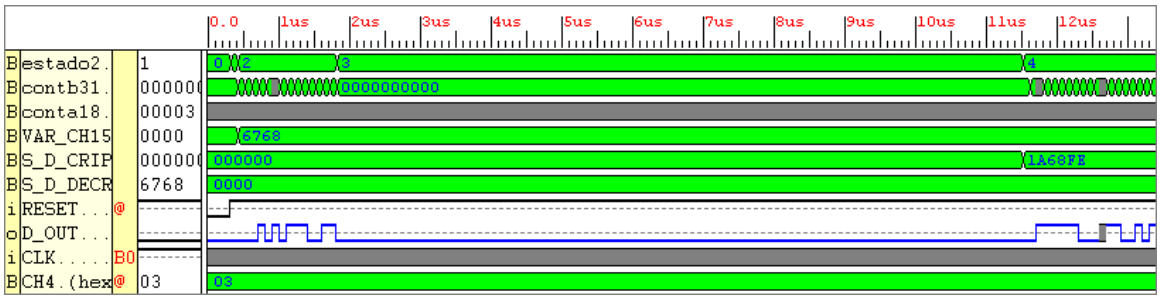

Figura 1 – Simulação do padrão PKCS#11 usando uma Máquina de estados Finito (MUZZI, 2005)

Em nosso projeto também implementamos o algoritmo RSA em hardware, em FPGA. Além disso, o nosso módulo RSA em hardware é parametrizado, permitindo mudar facilmente o tamanho da chave.

A figura 2 apresenta o impacto que o tamanho em bits das chaves do algoritmo RSA tem na ocupação do FPGA. Como se esperava, aumentando o tamanho (em bits) do algoritmo RSA, a ocupação (medida através de slice no FPGA), aumenta. Em nosso projeto foi possível verificar que mesmo a chave do algoritmo RSA com 512 bits junto com a máquina de estado na especificação do padrão PKCS#11, verifica-se que o máximo de ocupação do FPGA foi de 17,96% sendo possível usar o espaço não ocupado em outros projetos futuros, como por exemplo um processador de rede.

Nós verificamos o impacto de ter tamanhos diferentes de chaves do algoritmo de criptografia RSA, sendo usado chave 24 bits, 56 bits, 128 bits, 256 bits e 512 bits. Quanto maior o número de bits usados na chave, maior será a ocupação do FPGA.

Esta seção apresenta os resultados obtidos usando a criptografia RSA em hardware. Foi utilizado para análise dos resultados um computador Pentium IV 1.6 GHZ com 128 de memória RAM e sistema operacional Windows 2000 e o Xilinx 6.2i.

O módulo de sintaxe retorna dados que são muito importantes para o levantamento de resultados, desde a ocupação do código RSA no FPGA até o número de IOBs, Flip Flop, LUTs e a velocidade máxima, medida em MHZ, obtidos após prototipação em FPGAs.

Na figura 3 pode-se visualizar a foto de um teste real usando computador, osciloscópio e FPGA no projeto que recebe o dado do FPGA pela porta serial do computador. Primeiro é enviado o dado original escolhido, por exemplo dois bytes que corresponde às letras “DE” logo após é enviado o dado criptografado e depois o dado decriptografado.

Foram criadas três versões da máquina de estados finito - FSM, a versão 3.1, 6.1 e 6.2. A versão 3.1 foi implementada na ferramenta Xilinx versão 3.1 e simulada usando a ferramenta de síntese FPGA Express. A versão 6.1 foi

implementada no Xilinx 6.2 e finalmente a versão 6.2 foi também implementada no Xilinx 6.2.

Figura 3 - Comunicação real com o PKCS#11 em hardware.

6. CONCLUSÃO

A criação de um protótipo com módulo PKCS#11 para segurança em processadores de rede é importante, já que o PKCS#11 é um padrão de segurança criado para criptografia baseada em token, podendo ser usado em software como em hardware.

A implementação do protótipo com o padrão PKCS#11 foi feita em VHDL e prototipado em hardware usando FPGA, seguindo uma implementação baseada em máquinas de estado finito. Além da análise de resultados obtidos usando criptografia em hardware, foi possível o envio de um dado criptografado usando o padrão PKCS#11 para outro computador usando um FPGA através de comunicação serial.

A máquina de estado implementada em hardware, chamada de FSM 6.2, usa o algoritmo de criptografia RSA, que também foi implementado em FPGAs, podendo ser usado qualquer um outro algoritmo de criptografia, não precisa ser específico o RSA. O importante neste projeto foi mostrar que o padrão de criptografia PKCS#11 pode ser implementado em hardware e não somente em software, enfatizando nos recursos utilizados em um determinado FPGA.

O algoritmo RSA em C foi implementado usando 24 bits e 56 bits, não foi possível aumentar o número de bits devido às variáveis em C não aceitarem tamanho maior que 32 bits, dessa forma não foi possível implementar o RSA de uma maneira simples em C com 128 bits, 256 bits e 512 bits e 1024 bits.

Na implementação usando VHDL foi possível implementar o RSA com 56 bits, 128 bits, 256 bits e 512 bits. Não foi possível implementar 1024 bits devido a

que em nossa implementação usou-se um vetor de 1024 bits, e a ferramenta Xilinx 6.2 tem restrições para vetores deste tamanho.

Para implementação em software usando o padrão PKCS#11 é possível usar uma biblioteca em C chamada cryptoki, em hardware não foi possível usar essa biblioteca sendo necessário abstrair o padrão PKCS#11 para a realização da implementação em hardware usando FPGA.

Foi usado o FPGA Spartan2E, no qual foi possível a realização de teste enviando um dado criptografado para outro computador usando uma comunicação serial e o software chamado hyperterminal do windows para verificar o dado que computador receptor recebeu. Como trabalhos futuros sugerimos implementar o algoritmo RSA com chaves maiores, e inserir a nossa solução em um processador de rede, também prototipado em FPGAs.

7. REFERÊNCIAS BIBLIOGRÁFICAS

CHIARAMONTE, BARROS, R. C, "Implementação e teste em Hardware e Software de Sistemas Criptográficos, 2003.

MORENO 2003, MORENO, E. et al. Projeto, Desempenho e Aplicações de Sistemas Digitais em Circuitos Programáveis (FPGAs). Bless, 2003.

MORENO 2005, MORENO, David Edward., PEREIRA, Fábio Dacêncio., CHIARAMONTE, Rodolfo Barros., Criptografia em Software e Hardware. Novatec, 2005.

RSA 2002, RSA Labs. Public Key Criptography Standards (PKCS). Version 2.1 - 2002, disponível em http://www.rsasecurity.com/rsalabs/pkcs/pkcs-1/index.html