DOI: http://dx.doi.org/10.29069/forscience.2014v2n2.e119

DESENVOLVIMENTO DE HARDWARE RECONFIGURÁVEL DE CRIPTOGRAFIA ASSIMÉTRICA

Otávio S. M. Gomes[1]
João Paulo F. C. César[2]

RESUMO

Este artigo apresenta o resultado parcial do desenvolvimento de uma interface de *hardware* reconfigurável para criptografia assimétrica que permite a troca segura de dados. *Hardwares* reconfiguráveis permitem o desenvolvimento deste tipo de dispositivo com segurança e flexibilidade e possibilitam a mudança de características no projeto com baixo custo e de forma rápida.

Palavras-chave: Criptografia. Hardware. ElGamal. FPGA. Segurança.

DEVELOPMENT OF AN ASYMMETRIC CRYPTOGRAPHY RECONFIGURABLE HARWADRE

ABSTRACT

This paper presents some conclusions and choices about the development of an asymmetric cryptography reconfigurable hardware interface to allow a safe data communication. Reconfigurable hardwares allows the development of this kind of device with safety and flexibility, and offer the possibility to change some features with low cost and in a fast way.

Keywords: Cryptography. Hardware. ElGamal. FPGAs. Security.

1 INTRODUÇÃO

A criptografia é usada como uma técnica de transformação de dados, segundo um código, ou algoritmo, para que eles se tornem ininteligíveis, a não ser para quem possui a chave do código (TERADA, 2000).

Existem duas técnicas básicas usadas na criptografia: transposição e substituição. A primeira é feita trocando as letras de posição em uma frase, a segunda substitui, por exemplo, letras em uma frase por outras letras (SINGH, 2011).

[1] Doutorando em Engenharia Elétrica pela Universidade Federal de Itajubá (UNIFEI), mestrado em Engenharia Elétrica pela UNIFEI e docente efetivo do Instituto Federal de Minas Gerais – Campus Formiga. E-mail: otavio.gomes@ifmg.edu.br.
[2] Graduando em Ciência da Computação pelo IFMG – Campus Formiga. E-mail: joaopaulofcc@gmail.com.

A criptografia pode ser classificada, também, quanto ao número de chaves utilizadas no processo de cifragem/decifragem, como simétrica e assimétrica. Na criptografia simétrica, emissor e destinatário compartilham a mesma chave, combinada previamente, para comunicação. Já a criptografia assimétrica foi desenvolvida com o intuito de suprir a grande desvantagem do modelo simétrico, por conseguir distribuir de forma segura e eficiente a chave (SINGH, 2011). Para isso, são utilizadas duas chaves, uma pública, propriedade única de um ente e divulgada livremente, e uma chave privada, conhecida somente pelo remetente e utilizada para decifrar a mensagem codificada com sua chave pública.

Criptografia em *hardware* é mais segura que a desenvolvida em *software* devido à dificuldade de se quebrar a chave e/ou descobrir como esse foi implementado. Os dispositivos reconfiguráveis permitem o desenvolvimento deste *hardware* de maneira segura e com uma grande flexibilidade (MATSUI, 1994; MORENO; PEREIRA; CHIARAMONTE, 2005).

Para desenvolver o algoritmo de ElGamal foi utilizado o teste probabilístico de Miller-Rabin para geração de números primos, a pseudoaleatoriedade será implementada seguindo o algoritmo *Linear Feedback Shift Register* (LFSR).

2 NÚMEROS PRIMOS

Um número é primo se possui apenas e exatamente dois divisores distintos, são eles ± 1 e $\pm n$. Números inteiros $n > 1$ que possuem divisores diferentes de ± 1 e $\pm n$ são chamados de números compostos. Segundo o Teorema Fundamental da Aritmética todo número inteiro pode ser decomposto de forma única em produtos de números primos compostos (CHAGAS, 2009).

2.1 O problema da fatoração e o problema primo

Encontrar um algoritmo que verifique a primalidade de um número não leva imediatamente a encontrar um algoritmo que fatore um número com mesma complexidade. (CHAGAS, 2009). São exemplos de testes de primalidade: método de divisão por tentativa, crivo de Eratóstenes, Teste de Primalidade de Fermat, Teste de Lucas-Lehmer, Teste de Solovay-Strassen e o Teste de Miller-Rabin.

2.2 Teste de Miller-Rabin

É um teste probabilístico criado em 1976 por G. L. Miller e modificado por M.O. Rabin (FARIA; SERCONEK, 2008). O funcionamento do algoritmo é mostrado a seguir.

- Dado um inteiro ímpar n a ser testado, deve-se escrever o número $n-1$ na forma $2^s.d$, onde s é um inteiro qualquer e d um inteiro ímpar;
- Escolhe-se um valor $a < n$ aleatório;
- Executa-se o primeiro teste $a^d \equiv 1(mod\ n)$;
- Executa-se o segundo teste $a^{2^i d} \equiv -1(mod\ n)$;

Caso qualquer um dos dois testes mostrados seja verdadeiro, o número n é declarado primo com uma probabilidade, assim, deve-se executar o algoritmo mais vezes para aumentar a confiança nesse resultado. Caso nenhum teste seja verdadeiro, o algoritmo retorna com certeza que n é composto.

3 NUMEROS ALEATÓRIOS

Números aleatórios são utilizados em diversos sistemas presentes no nosso cotidiano, eles fazem parte de simulações, tomada de decisões, jogos e entretenimento. Devido à dificuldade de gerar números realmente aleatórios, é utilizado o conceito de números pseudoaleatórios, que são gerados por algoritmos não determinísticos como o *Linear Congruential Generators* (LCG), *Mersenne Twister* e *Linear Feedback Shif Register* (LFSR).

3.1 LFSR

Consiste em um registrador de deslocamento capaz de gerar uma sequência de $2^n - 1$ *bits* pseudoaleatórios. Seu funcionamento é baseado em um polinômio primitivo de grau n (PINHEIRO, [20-?].). Sua montagem é realizada por meio de n *flip-flops*, entre a saída de um registrador *(flip-flop)* e a entrada de uma porta Ou-Exclusivo existe uma ligação chamada de *tap*. O período da sequência de um LFSR depende da sua quantidade de *flip-flops* e também do número e da posição dos taps.

4 O ALGORITMO DE ELGAMAL

Criado em 1985 por Taher ElGamal (ELGAMAL, 1985) sua segurança é baseada no Problema do Logaritmo Discreto (MOLLIN, 2006).

4.1 Geração de chaves

Deve-se escolher um número primo grande p e um gerador $\propto$ do grupo multiplicativo $\mathbb{Z}_p^*$. Em seguida seleciona-se aleatoriamente um número natural $a < p-1$ e calcula-se o valor da expressão $\propto^a \pmod p$. Após esses passos é encontrada a chave pública dada pela 3-upla $(p, \propto, \propto^a)$ e pela chave privada dada pelo valor de a.

4.2 Cifragem

O emissor *A* da mensagem deve obter a chave pública do destinatário *B*. Após esse passo, *A* deve converter os caracteres de sua mensagem em números m, esses contidos no intervalo de 0 e $p-1$, caso $p \geq 256$ pode-se utilizar a tabela ASCII estendida.

Escolhe-se aleatoriamente um número natural $b < p-1$, esse valor é calculado para cada caractere a ser enviado. Recomenda-se usar um b distinto para cada m a ser criptografado (TERADA, 2008). Assim, para cada caractere m convertido acima, deve-se calcular os seguintes valores: $\Omega \equiv \propto^b \pmod p$ e $\theta \equiv m(\propto^a)^b \pmod p$. Em seguida o emissor envia ao destinatário a cifragem $c = (\Omega, \theta)$ correspondente ao caractere m.

4.3 Decifragem

Quando *B* recebe a mensagem criptografada de *A*, para decifrá-la ele deverá, utilizando sua chave privada, calcular primeiramente $\varphi = \Omega^{p-1-a} \pmod p$. Feito esse cálculo, ele deverá em seguida decifrar m calculando $m = \varphi\theta \pmod p$. Dessa forma, ele terá obtido um caractere m enviado por *A* através de um meio inseguro de forma segura.

5 CONCLUSÃO

A implementação dos algoritmos de criptografia e geração de números aleatórios em *software*, desenvolvidos em linguagem C, foi realizada com sucesso e representa a primeira parte deste projeto, que está em andamento. O próximo passo será a implementação em *hardware* desses algoritmos, juntamente com o algoritmo LFSR. Com a finalização dos protótipos em FPGA, serão avaliados o desempenho e corretude dos algoritmos em *hardware* por meio de testes em ferramentas de medição.

AGRADECIMENTO

Os autores agradecem o Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – IFMG pelo apoio financeiro por meio do projeto de pesquisa aprovado no Edital: 156/2013.

REFERÊNCIAS

CHAGAS, A. B. **Testes de primalidade**: uma visão computacional. 2009. 46 f. TCC (Graduação em Ciência da Computação) - Centro de Informática, Universidade Federal de Pernambuco, Recife, 2009. Disponível em: <http://www.cin.ufpe.br/~tg/2009-2/abc.pdf>. Acesso em: 8 nov. 2014.

ELGAMAL, T. A public key cryptosystem and a signature scheme based on discrete logarithms. In: BLAKLEY, G. R.; CHAUM, D. (Eds.). **Advances in cryptology - CRYPTO'84**. Berlin: Springer-Verlag, 1985. p. 10-18.

FARIA, M. A. de; SERCONEK, S. Mini curso "Números primos: Testes de primalidade e aplicações". In: SEMANA DO IME, 23., 2008, Goiânia. **Anais...** Goiânia: Ime, 2008. p. 1 - 18. Disponível em: <http://semanadoime.mat.ufg.br/up/34/o/min_Cida.pdf>. Acesso em: 9 nov. 2014.

MATSUI, M. Linear cryptanalysis method for DES cipher. In: HELLESETH, T. (Ed.). **Advances in Cryptology - EUROCRYPT'93**. Berlin: Springer-Verlag, , 1994. p. 386-397.

MOLLIN, R. A. **An introduction to cryptography**. 2nd. Boca Raton: CRC Press, 2006.

MORENO, E. D.; PEREIRA, F. D.; CHIARAMONTE, R. B. **Criptografia em software e hardware.** São Paulo: Novatec, 2005. 288 p.

PINHEIRO, G. R. V. **Linear feedback shift register (LFSR)**. Universidade do Estado do Rio de Janeiro. [20-?]. Disponível em: <http://www.lee.eng.uerj.br/~gil/redesII/Linear_Feedback Shift_Register_(LFSR).pdf>. Acesso em: 15 nov. 2014.

SINGH, S. **O Livro dos códigos.** 9. ed. Rio de Janeiro: Record, 2011. 448 p.

TERADA, R. **Segurança de dados:** criptografia em redes de computador. São Paulo: Blucher, 2000. 248 p.

Recebido em: 16 / 11 / 2014

Aprovado em: 28 / 11 / 2014

Publicado em: 26/01/2015