Um gerador de bits pseudo-aleatórios seguro baseado em curvas elípticas

Afonso Comba de Araujo Neto, Raul Fernando Weber

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

afonso@cpd.ufrgs.br, weber@inf.ufrgs.br

Abstract. *This paper presents the proposal of a cryptographically strong pseudorandom bit generator with a hybrid architecture based on both modular exponentiation and elliptic scalar product over a pair of elliptic curves. Its security is presented in theoretical terms. This proposal is part of a research about the viability of a stream cipher based on elliptic curves.*

Resumo. *Este artigo apresenta a proposta de um gerador de bits pseudo-aleatórios criptograficamente seguros com uma arquitetura híbrida que utiliza exponenciação em módulo e produto escalar em um par de curvas elípticas. Sua segurança é apresentada em termos teóricos. Esta proposta faz parte de uma pesquisa sobre a viabilidade de um cifrador de fluxo baseado em curvas elípticas.*

1. Introdução

Curvas elípticas têm se tornado um assunto cada vez mais presente no campo da criptografia. Atualmente, tem-se que a principal forma de utilização de curvas elípticas na criptografia se dá através de criptossistemas de chave pública. Isso aconteceu pois tanto a comunidade acadêmica quanto as empresas de tecnologia aceitaram definitivamente as vantagens dos criptossistemas elípticos sobre os sistemas baseados no RSA.

É possível, entretanto, utilizar os sistemas elípticos como base para outras técnicas criptográficas. Uma destas técnicas é a *geração de seqüências pseudo-aleatórias*, que tem utilidade não só na criptografia, mas também em diversas outras áreas. De fato, nos últimos tempos, tem-se dado muito menos importância à geração de seqüências pseudo-aleatórias seguras do que se deveria. Todos os algoritmos criptográficos dependem fundamentalmente da qualidade destes geradores, e, conseqüentemente, não se deve utilizar os geradores lineares congruentes fornecidos diretamente por bibliotecas padrão, que são extremamente inseguros [Boyar 1989]. Uma prova disso é a técnica desenvolvida por Bellare, Goldwasser e Micciancio que permite obter a chave secreta do algoritmo DSA (*Digital Signature Algorithm*) caso os valores aleatórios requeridos pelo mesmo sejam gerados utilizando-se um gerador linear congruente [Bellare et al. 1997].

A teoria acerca de geradores de seqüências pseudo-aleatórias criptograficamente seguras remonta ao início década de 80, e seu princípio se atribui ao artigo de Andrew C. Yao [Yao 1982]. Neste artigo se estabelece que problemas difíceis como a fatoração em componentes primos, o logaritmo discreto e o resíduo quadrático podem ser utilizados na arquitetura de geradores de bits pseudo-aleatórios criptograficamente seguros

(ou *Cryptographically Strong Pseudorandom Bit Generator*, CSPRBG). O primeiro CSPRBG prático que se tem notícia é o gerador Blum-Micali, baseado na dificuldade de se computar logaritmos discretos e gera um bit por iteração [Blum and Micali 1982].

Kaliski em [B. S. Kaliski 1988] desenvolveu um gerador criptograficamente seguro, mas ineficiente, baseado no problema do logaritmo discreto elíptico (*Elliptic Curve Discrete Logarithm Problem*, ECDLP). Enquanto os geradores baseados em logaritmo discreto convencional fornecem $\Theta(\frac{\log p}{2})$ bits[1] por iteração [Goldreich and Rosen 2000], ninguém sabe como obter mais que um número logarítmico de bits de um gerador baseado em curvas elípticas. Este trabalho apresenta uma proposta híbrida, que utiliza parte da construção de Kaliski e fornece $\lfloor \log p - c \log \log p \rfloor$ bits por iteração (para alguma constante c), enquanto mantém a ordem de segurança fornecida por logaritmos elípticos.

2. Curvas elípticas e o gerador de Kaliski

Uma curva elíptica é uma equação com a seguinte forma geral $E : y^2 + Axy + By = x^3 + Cx^2 + Dx + E$ definida sobre um corpo qualquer $\mathbb{K}$, com A, B, C, D e $E \in \mathbb{K}$. O corpo $\mathbb{K}$ é um conjunto de elementos, finito ou não, associado às operações de multiplicação e adição, que devem possuir inversos aditivo e multiplicativo e respectivas identidades. Freqüentemente se utiliza o conjunto dos inteiros módulo um número primo p ($\mathbb{K} = \mathbb{F}_p$), ou então um conjunto de polinômios módulo um polinômio irredutível ($\mathbb{K} = \mathbb{F}_{p^\alpha}$).

Seja $E(\mathbb{K})$ o conjunto de pontos que satisfazem a curva elíptica E sobre $\mathbb{K}$. Adiciona-se ao conjunto o ponto extra chamado *ponto no infinito*, denotado por $\mathcal{O}$. É possível definir uma operação chamada *adição de pontos*, de tal forma que quaisquer dois pontos $\mathcal{P}, \mathcal{Q} \in E(\mathbb{K})$ somados, resultam em um outro ponto do conjunto. O ponto no infinito é a identidade do sistema, sendo então $\mathcal{P} + \mathcal{O} = \mathcal{P}$ e $\mathcal{P} - \mathcal{P} = \mathcal{O}$. Naturalmente, é possível obter a soma de um ponto $\mathcal{P}$ com ele mesmo: $\mathcal{P} + \mathcal{P} = 2\mathcal{P}$, que também pode ser visto como a sua multiplicação pelo escalar 2. Seguindo essa idéia, pode-se definir o *produto escalar elíptico*, como a soma $k\mathcal{P} = \mathcal{P} + \mathcal{P} + ... + \mathcal{P}$ (k vezes). A soma elíptica junto a este conjunto de pontos formam um *grupo*. O ECDLP é então definido como: dados os pontos $\mathcal{P}, \mathcal{Q} \in E(\mathbb{K})$ tal que $\mathcal{Q} = k\mathcal{P}$, calcular o valor k.

Define-se a ordem N do ponto $\mathcal{G}$ como o valor escalar tal que $\mathcal{O} = N\mathcal{G}$. Em outras palavras, N é o número de pontos distintos na seqüência $\{\mathcal{G}, 2\mathcal{G}...(N-1)\mathcal{G}, \mathcal{O}\}$. O ponto $\mathcal{G}$ é dito o ponto gerador desta seqüência, que é um *subgrupo* cíclico de E. Caso a curva tenha ordem finita prima, o subgrupo de $\mathcal{G}$ é igual ao conjunto de pontos da curva.

2.1. Um CSPRBG baseado em curvas elípticas

Burton Kaliski apresenta um CSPRBG cuja construção utiliza um par de curvas, denominadas um *twisted pair*. Essas curvas são relacionadas através de seus coeficientes da seguinte forma. Seja p um número primo, $\mathbb{F}_p$ um corpo finito e E uma curva elíptica do tipo $y^2 = t^3 + At + B$ sobre $\mathbb{F}_p$. Seja γ um não-resíduo quadrático em $\mathbb{F}_p$, ou seja, não existe solução para a equação $x^2 = \gamma \mod p$. Seja $A^{tw} = A\gamma^2$ e $B^{tw} = B\gamma^3$. O *twisted pair* de E é definido como $E^{tw} : y^2 = t^3 + A^{tw}t + B^{tw}$.

Kaliski mostra que essa construção apresenta as seguintes propriedades: para todo $s \in \mathbb{F}_p$, ou existe um ponto (x, y) com coordenada $x = s$ em $E(\mathbb{F}_p)$ ou então existe um

[1]Neste trabalho, todos os logaritmos são considerados base 2.

ponto com coordenada $x = s\gamma$ em $E^{tw}(\mathbb{F}_p)$. Além disso, quando o número de pontos das curvas é primo, quaisquer dois pontos não triviais $\mathcal{G} \in E(\mathbb{F}_p)$ e $\mathcal{G}^{tw} \in E^{tw}(\mathbb{F}_p)$ são geradores de ordem máxima, e a seguinte relação entre suas ordens é verdadeira: $N + N^{tw} = 2p+2$. Na época de sua tese, não se sabia como obter curvas com um número primo de pontos de forma eficiente. Hoje diversas técnicas para isso estão disponíveis.

Finalmente, cria-se uma bijeção entre os conjuntos de pontos das duas curvas e os valores inteiros $[0, 2p + 1]$. Simplificadamente, o gerador funciona da seguinte forma: dado $i \in \{0..2p + 1\}$, então se $i \leq N$, calcular $i\mathcal{G}$, caso contrário, calcular $(i - N)\mathcal{G}^{tw}$. Utilizando-se as coordenadas x e y do ponto resultado, e o valor γ, calcula-se um novo valor no intervalo $[0, 2p + 1]$ e o processo recomeça. Kaliski mostra também que os $\log\log\log p$ bits mais significativos de i, a cada iteração, são criptograficamente seguros.

3. Um CSPRBG híbrido baseado em curvas elípticas

Considere um corpo finito $\mathbb{F}_p$ tal que o valor $q = 2p + 1$ também seja primo. Então $\mathbb{F}_q$ também é um corpo. A proposta é bastante simples. Seja g um gerador do grupo multiplicativo $\mathbb{F}_q^*$. Seja $\chi : \{0, \ldots, 2p+1\} \rightarrow \{0, \ldots, 2p+1\}$ uma função definida como *uma* iteração do gerador de Kaliski, ou seja, uma função unidirecional bijetora entre os valores do intervalo $[0, 2p + 1]$ definida pelo produto escalar em um *twisted pair* sobre $\mathbb{F}_p$. Seja x_n uma semente uniformemente escolhida no intervalo $[0, 2p + 1)$. Dado algum fator constante de segurança c, o gerador possui a seguinte iteração:

$$x_{n+1} = g^{x_n} \mod q$$

fornecendo os $\lfloor \log p - c \log\log p \rfloor$ bits menos significativos do valor $\chi(x_{n+1})$.

É importante considerar que geradores baseados somente em logaritmos discretos são seguros somente sobre corpos com elementos da ordem de mil bits, sendo que as curvas elípticas são seguras sobre corpos com elementos da ordem de menos de duzentos bits. Considerando-se que os geradores com exponenciação tradicional são criptograficamente seguros quando se utiliza como expoente apenas metade dos bits de cada passo, estes geradores operam pelo menos quinhentos bits por iteração. No gerador proposto, mesmo sendo necessária uma exponenciação tradicional seguida de um produto escalar elíptico, as operações envolvem apenas em torno de quatrocentos bits.

4. Criptoanálise da proposta

Uma das virtudes da utilização de curvas elípticas na criptografia vem do desconhecimento de um algoritmo subexponencial para calcular logaritmos. O *index calculus* que é utilizado para calcular logaritmos sobre $\mathbb{F}_p^*$ não é aplicável a logaritmos elípticos[2]. Essa impossibilidade se deve à falta de uma estrutura bem definida nos elementos do grupo formado pelas curvas. Isso faz com que somente algoritmos genéricos, que não dependam dessa estrutura, sejam aplicáveis, e estes são sempre exponenciais. Na construção proposta, caso exista um mapeamento entre os valores $\chi(x_n)$ e x_n de forma que seja possível aplicar o *index calculus* diretamente sobre $\chi(x_n)$, então este seria um método subexponencial de se resolver o ECDLP no caso genérico, o que não se acha possível.

[2] A não ser no caso de curvas com grau de mergulho baixo, como as supersingulares. Entretanto, é fácil identificar e evitar estas curvas.

Um detalhe interessante é o fato de que, caso se descubra uma estrutura no grupo de pontos das curvas elípticas de forma a permitir um ataque subexponencial, então o interesse nas mesmas perde-se imediatamente. Isso é óbvio, na medida que os grupos multiplicativos são muito mais simples que os elípticos. Por este raciocínio, presumir que esta estrutura não exista é perfeitamente aceitável. Logo, é plausível assumir o seguinte resultado de C. P. Schnorr sobre grupos sem estrutura: *em qualquer grupo sem estrutura que possua ordem prima, qualquer fração de j bits que seja criptograficamente segura implica que todas as outras frações de j bits também o sejam* [Schnorr 1998]. Nestes termos, a prova de Kaliski de que os $\log\log\log p$ bits mais significativos do logaritmo são simultaneamente seguros implica que *todos* os bits do logaritmo de grupos elípticos com ordem prima sejam individualmente seguros. Em outras palavras, calcular o valor de qualquer bit de x_n no gerador proposto é tão difícil quanto calcular o logaritmo discreto elíptico completo. Na medida que o gerador não fornece nenhum bit do logaritmo como saída, então não é necessário que eles possuam segurança simultânea. É importante notar, entretanto, que uma informação sobre x_n é fornecida: sabe-se de qual das curvas do par de curvas o valor $\chi(x_n)$ foi calculado, e isso informa a ordem de grandeza do valor x_n. Mais especificamente, se x_n é maior ou menor que a ordem da primeira curva. Entretanto, sabe-se que os bits mais significativos da exponenciação em módulo são simultaneamente seguros, logo, esta informação não é suficiente para quebrar o sistema.

Resta entender porque os $\lfloor \log p - c\log\log p \rfloor$ bits menos significativos de cada valor $\chi(x_n)$ são polinomialmente indistinguíveis de uma seqüência aleatória. Esse fato advém da constatação de que tanto a exponenciação modular quanto a função χ formam *permutações unidirecionais*. Isso significa que, para sementes uniformemente escolhidas, todos os valores menores que $2p + 2$ são igualmente prováveis. Os $c\log\log p$ bits mais significativos são descartados, pois são claramente bits tendenciosos nesse intervalo, e a constante c, na prática, dependente da ordem do primo p e da segurança desejada. Logo, como todos os valores são igualmente prováveis então todos os bits são seguros.

Referências

B. S. Kaliski, J. (1988). Elliptic curves and cryptography: A pseudorandom bit generator and other tools. In *Phd Thesis*. MIT.

Bellare, M., Goldwasser, S., and Micciancio, D. (1997). "Pseudo-Random" number generation within cryptographic algorithms: The DSS case. In *CRYPTO '97*, pages 277–291, London, UK. Springer-Verlag.

Blum, M. and Micali, S. (1982). How to generate cryptographically strong sequences of pseudorandom bits. In *Proceedings of IEEE Symposium on Foundations of Computer Science*. IEEE.

Boyar, J. (1989). Inferring sequences produced by pseudo-random number generators. *J. ACM*, 36(1):129–141.

Goldreich, O. and Rosen, V. (2000). On the security of modular exponentiation with application to the construction of pseudorandom generators. *Cryptology ePrint Archive*.

Schnorr, C. P. (1998). Security of almost all discrete log bits. *ECCC*, (TR98-033).

Yao, A. C. (1982). Theory and applications of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science*, pages 80–91. IEEE.