Параллельный метод Полларда решения задачи дискретного логарифмирования с использованием детерминированной функции разбиения на множества

Качко Е.Г. Погребняк К.А.

Харьковский национальный университет радиоэлектроники

31 марта 2013 г.

Актуальность и цель исследований

1. Оценка стойкости криптографических примитивов заключается в выборе наиболее эффективного метода решения вычислительно сложной задачи и дальнейшем получении эмпирических характеристик выбранного метода.
2. Критерием эффективности методов криптоанализа, как правило, считается минимизация вычислительных ресурсов, то есть пространственно-временных показателей, и как результат удешевление процесса криптоанализа.
3. Однако, следует отметить что, оценивая стойкость, необходимо максимально точно учитывать современные возможности криптоаналитической системы. Под такими возможностями следует понимать использование кластерных систем, многоядерных и многопроцессорных компьютеров, а также облачных вычислений.
4. Таким образом, **актуальной задачей** является усовершенствование методов криптоанализа для эффективного использования в современных вычислительных системах с разными характеристиками.

Дискретное логарифмирование

Не ограничивая общности, в дальнейшем будет использоваться аддитивная форма записи.

Пусть задана абелева группа G, такая что $\#G = n$, где n – простое число. Зафиксируем образующий элемент группы P.

Определение

Для произвольного элемента группы $Q \in G$, такого что $Q = xP$ задача дискретного логарифмирования заключается в нахождении элемента $1 < x < n$.

Структура ρ-метода Полларда решения задачи дискретного логарифмирования

- Фактически, ρ-метод Полларда включает в себя:
  - алгоритм построения псевдослучайной последовательности;
  - алгоритм обнаружения коллизии.
- Повышение эффективности криптоанализа ρ-методом Полларда достигается за счет:
  - сокращения длины последовательности значений функции итерирования;
  - улучшения алгоритма обнаружения коллизии;
  - распараллеливания вычислений.

Последовательный ρ-метод Полларда

(Алгоритм итерирования элементов)

- Группа G представляется в виде объединения $G = S_1 \cup S_2 \cup S_3$, где S_i – произвольные множества приблизительно одинаковой мощности. Функция итерирования $f : G \to G$ определяется как

$$R_{i+1} = f(R_i) = \begin{cases} Q + R_i, & R_i \in S_1 \\ 2R_i, & R_i \in S_2 \\ P + R_i, & R_i \in S_3 \end{cases} \quad (1)$$

- Так как группа G – конечна, то последовательность $\{R_i\}_{i=0}^{\infty}$ – периодическая. Таким образом, найдутся два наименьших натуральных числа t и l, таких что $R_t = R_{t+l}$. Фактически, l – означает длину периода последовательности.

Последовательный ρ-метод Полларда

(Алгоритм детектирования цикла Флойда)

- Идея алгоритма детектирования цикла Флойда заключается в нахождении индекса i_0, такого что $R_{i_0} = R_{2i_0}$, $i_0 \leq t + l$ при произвольно фиксированном начальном значении R_0.
- На каждой итерации производится сравнение R_i и R_{2i} для $0 < i \leq t + l$ пока не будет обнаружена коллизия $R_{i_0} = R_{2i_0}$, где

$$\begin{aligned} R_i &= f(R_{i-1}) \\ R_{2i} &= f(f(R_{i-1})) \end{aligned} \tag{2}$$

- Учитывая, что

$$\begin{aligned} R_{i_0} &= a_{i_0} P + b_{i_0} Q \\ R_{2i_0} &= a_{2i_0} P + b_{2i_0} Q \end{aligned} \tag{3}$$

можно вычислить

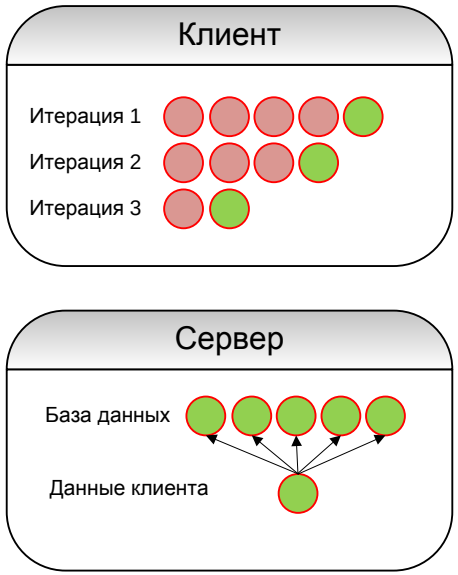
$$x = \frac{a_{2i_0} - a_{i_0}}{b_{i_0} - b_{2i_0}} \tag{4}$$

ρ-метод Полларда для систем с распределенной памятью

(Ооршот, Вейнер)

Идея метода состоит в разделении итерирования точек между клиентскими рабочими станциями и поиска коллизии сервером.

1. Сервер определяет общесистемные параметры, некоторое подмножество $D \subset G$ и выполняет инициализацию рабочих станций.
2. Клиентская рабочая станция S_i строит последовательность точек $\{R_{ij}\}_{j=0}^{m} \subset D$ и отправляет поэлементно точки на сервер.
3. Если точка не содержится в базе данных, сервер добавляет точку в базу данных, иначе вычисляет значение дискретного логарифма.



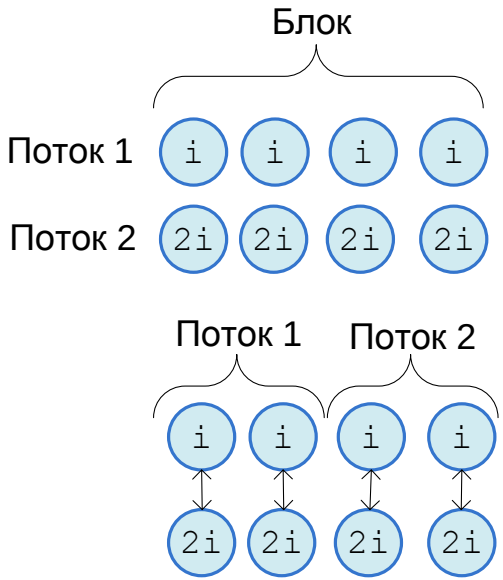
ρ-метод Полларда для систем с общей памятью

Идея метода состоит в распараллеливании отдельно функции итерирования и алгоритма обнаружения коллизии.

- Функция итерирования распараллеливается на этапе вычисления последовательностей $\{R_i\}_{i=0}^{m}$ и $\{R_{2i}\}_{i=0}^{m}$. Это достигается организацией вычислений последовательностей вида

$$\{R_{iw+j}\}_{i=0}^{l} \text{ и } \{R_{2(iw+j)}\}_{i=0}^{l} \tag{5}$$

где w – это размер блока вычислений, $0 \leq j < w$, $l = \left\lceil \frac{m}{w} \right\rceil$.



- Результатом выполнения функции итерирования точек являются два множества точек $\{R_i\}_{i=0}^{w}$ и $\{R_{2i}\}_{i=0}^{w}$, сравнение которых осуществляется по блокам, то есть

$$\begin{aligned} R_i &= R_{2i}, 1 \leq i \leq \frac{w}{2} \\ R_i &= R_{2i}, \frac{w}{2} < i \leq w \end{aligned} \tag{6}$$

ρ-метод Полларда с использованием детерминированной функции разбиения на множества

- Определим детерминированную функцию $\xi : G \to \{1, \ldots N\}$ разбиения на множества как

$$j = \xi(R_i) = i \mod N + 1. \tag{7}$$

- Такая функция разбиения абелевой группы на множества позволяет сделать предсказание траектории перехода и построить композиционную функцию.

Утверждение Пусть задана детерминированная функция разбиения группы на множества $\xi(R_i) = i \mod N + 1$, тогда вычисление R_{2i} может быть представлено как

$$R_{2i} = g(S_{i-1}), \tag{8}$$

где $g : G \to G$, $S_{i-1} = R_{2(i-1)}$.

Комбинированный ρ-метод Полларда

- ρ-метод Полларда для систем с распределенной памятью может быть расширен для использования на многоядерных рабочих станциях.
- Идея метода состоит в том, что каждая клиентская рабочая станция S_i строит последовательность точек $\{R_{ij}\}_{j=0}^{w} \cap D$ и отправляет его серверу.
- Проверка на принадлежность подмножеству осуществляется в параллельном режиме:

$$\begin{aligned} R_{ij} \in D, 1 \leq i \leq \frac{w}{2} \\ R_{ij} \in D, \frac{w}{2} < i \leq w. \end{aligned} \tag{9}$$

- Сервер добавляет точки $\{R_{ij}\}_{j=0}^{w}$ в базу данных до тех пор, пока не найдет уже существующую точку.

Теоретические оценки времени выполнения и требуемой памяти

(Для двуядерных рабочих станций)

№	Методы Полларда	Время выполнения	Количество памяти
1	Классический метод	$t\sqrt{\frac{\pi n}{2}}$	1
2	Параллельный метод с общей памятью	$\frac{t}{2}\sqrt{\frac{\pi n}{2}}$	1
3	Параллельный метод с разделенной памятью	$t\left(\sqrt{\frac{\pi n}{2}}+\frac{1}{\theta}\right)$	$\theta\sqrt{\frac{\pi n}{2}}$
4	Комбинированный метод	$\frac{t}{2}\left(\sqrt{\frac{\pi n}{2}}+\frac{1}{\theta}\right)$	$\theta\sqrt{\frac{\pi n}{2}}$

Временные показатели для последовательного и параллельного методов Полларда

(для систем с общей памятью)

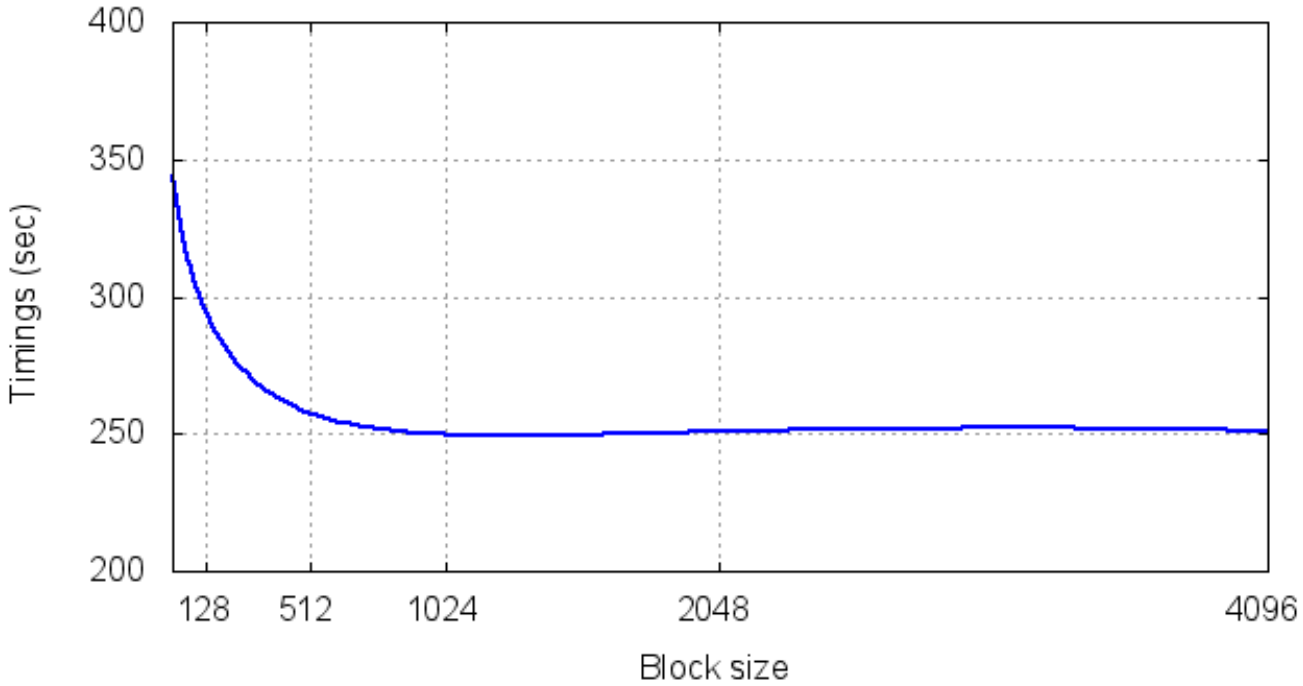
№	Методы Полларда	δ_t (сек)	
		21 бит	**27 бит**
1	Классический метод	9.59973	354.191
2	Параллельный метод (w=1)	8.50658	344.427
3	Параллельный метод (w=4)	7.22392	308.106
4	Параллельный метод (w=8)	7.10515	278.301
5	Параллельный метод (w=128)	6.95188	246.382
6	Параллельный метод (w=512)	6.93911	245.298
7	Параллельный метод (w=1024)	6.98354	245.65
8	Параллельный метод (w=2048)	7.08218	256.072
9	Параллельный метод (w=4096)	7.21804	251.537

Временные показатели для последовательного и параллельного методов Полларда

(для систем с общей памятью с использованием детерминированной функции разбиения на множества)

№	Методы Полларда	δ_t (сек)	
		21 бит	**27 бит**
1	Классический метод	12.4141	1069.49
2	Параллельный метод (w=1)	9.6856	890.812
3	Параллельный метод (w=4)	8.70438	765.313
4	Параллельный метод (w=8)	8.45161	786.541
5	Параллельный метод (w=128)	8.21501	810.267
6	Параллельный метод (w=512)	8.17755	726.259
7	Параллельный метод (w=1024)	8.1422	706.71
8	Параллельный метод (w=2048)	8.1446	702
9	Параллельный метод (w=4096)	8.21867	706.509

Зависимость временных показателей от размера блока



Временные показатели избыточных операций метода Полларда

№	Методы Полларда	Δ_t (мк)			
		21 бит		27 бит	
		I	C	I	C
1	Параллельный метод (w=4)	32	4,8	32,6	4,89
2	Параллельный метод (w=8)	64,5	9,4	87,1	9,24
3	Параллельный метод (w=128)	1000	170	1031	139
4	Параллельный метод (w=512)	4041	545	4050	548
5	Параллельный метод (w=1024)	8047	1115	8169	1148
6	Параллельный метод (w=2048)	16162	2411	16488	2477
7	Параллельный метод (w=4096)	32582	4881	33039	4981

Выводы

- В работе предложен метод распараллеливания алгоритма Полларда решения задачи дискретного логарифмирования в мультипликативной группе поля Галуа и в группе точек эллиптической кривой для систем с общей памятью с использованием детерминированной функции разбиения на множества.
- Проанализированы известные функции итерирования точек в алгоритме Полларда и построен обобщенный метод Полларда для произвольной функции итерирования и систем с общей памятью с использованием детерминированной функции разбиения на множества.
- Предложенный подход к распараллеливанию алгоритма Полларда без использования детерминированной функции разбиения на множества позволяет снизить время вычислений на 30%, а с использованием – на 35%.
- В дальнейшем планируется провести моделирование для комбинированного метода и для различных итеративных функций, обобщить полученные результаты на случай произвольного числа ядер и на кривые с большей битовой длиной порядка подгруппы.
- Отметим также, что анализировался только один алгоритм обнаружения цикла, а именно алгоритм Флойда. Необходимо также проанализировать альтернативные алгоритмы, например, алгоритм Брента.