A SHORT OVERVIEW OF THE DISCRETE LOGARITHM PROBLEM

Jorge Nakahara Jr

Katholieke Universiteit Leuven
Department of Electrical Engineering - ESAT
SISTA/COSIC Research Group

Outline

- Description of the Discrete Logarithm Problem (DLP)
- Motivations (some applications of the DLP)
- Historical Perspective
- Algorithms for Solving the DLP
- Generalizations of the DLP
- Conclusions

Description of the DLP

Let G be a finite cyclic group of order n, and g a generator of G. For any given integer $x \in [0, n-1]$, the modular (or discrete) exponentiation g^x is quite efficient to perform on G, but given an element $y \in G$, the inverse operation, called the **Discrete Logarithm (DL)** of y in G which is to determine the smallest non-negative integer x such that

- $y = g^x$ (in case G is written *multiplicatively*), or
- $y = xg$ (in case G is written *additively*)

is not usually trivial. This is the Discrete Logarithm Problem (DLP).

Notation: the discrete logarithm of y to the base g is denoted $x = \log_g y$. (x is also known as the *index* of y in G).

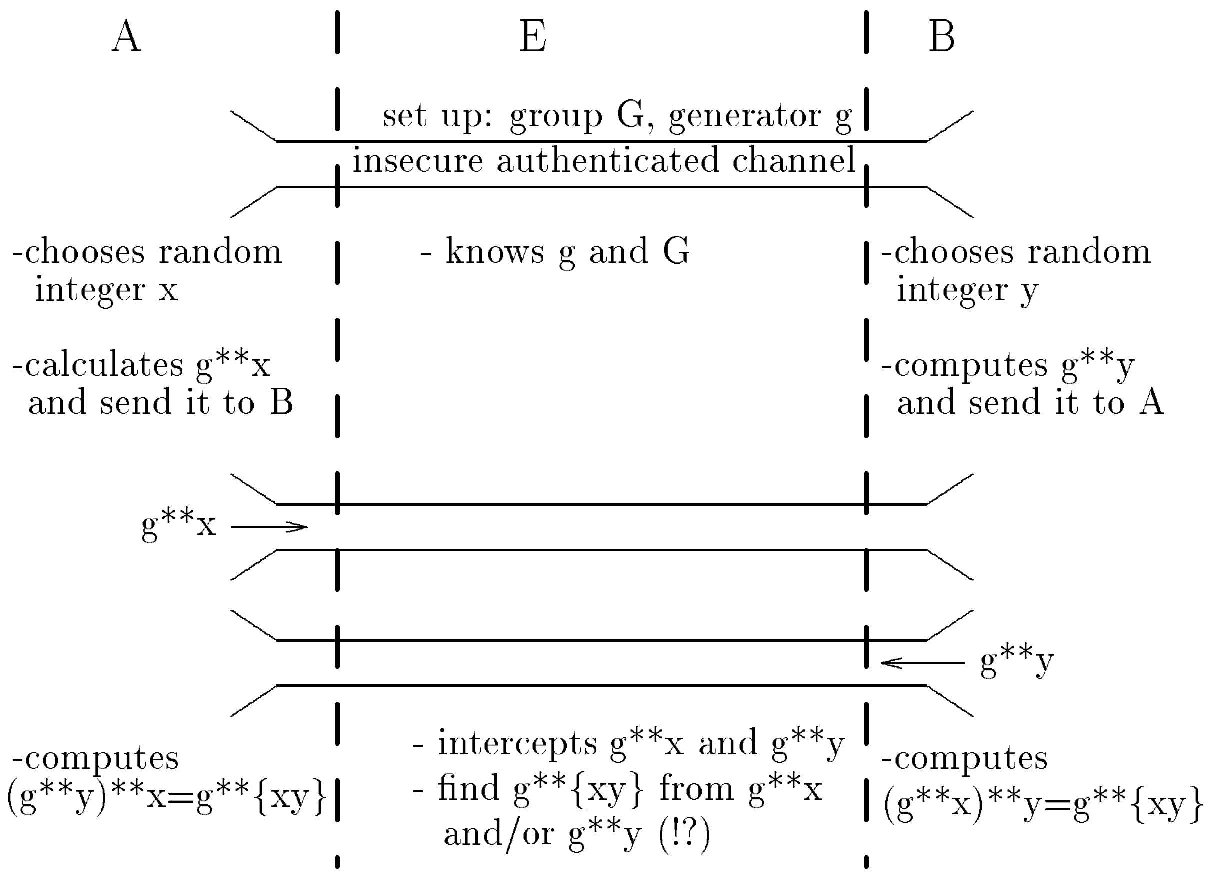
The presumed intractability of the DLP, for appropriate choices of G, in contrast to the relative efficiency of calculating the discrete exponentiation, has made the DLP a basic building block of many cryptographic applications, including public-key encryption algorithms, digital signature schemes, and key agreement protocols.

Some Motivations (for studying the DLP)

Some cryptographic applications based on the DLP :

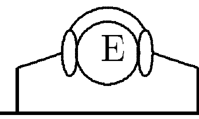
- Key-agreement protocol (Diffie-Hellman-1976)
- Public-key algorithm (ElGamal-1984)
- Identification and signatures for Smart Cards (Schnorr-1989)
- Identification schemes (McCurley-1990)
- Zero-knowledge schemes (Desmedt/Burmester-1991)
- Digital signature scheme (ElGamal-1984, Hardjano/Zheng-1992)
- Blind signatures (Camenisch/Piveteau/Stadler-1994)
- Pseudo-random number generators (Sundaram-1998)

Basic Diffie-Hellman Key Agreement Protocol



Basic ElGamal Signature Scheme

A



B

insecure authenticated channel

A and B agree on a common large prime p, and a generator g of Z*p

Key Generation

-choose random integer x, in the range [1,p-2]

-compute y=g**x mod p

-A's public key is (p,g,y)

-A's private key is x

Signature Generation

-to sign a message m

-select a random integer k in [1,p-2] with gcd(k,p-1)=1

-compute the hash h(m) of m; k**(-1) mod (p-1);
r=g**k mod p and
s=k**(-1){h(m)-xr} mod(p-1)

-A's signature for m is (r,s)

(r,s) →

(Passive) Eavesdropping

- E knows p,g, m, r and s

-if E wants to forge a signature (r',s') for a new message m'

-with r'=r, then E has to solve g**{h(m')}=y**r.r**s' mod p for s' (DLP instance)

-with s'=s, then E has to solve g**{h(m')}=y**r'.(r')**s mod p

-otherwise E has to solve g**{h(m')}=y**r'.(r')**s' mod p

Signature Verification

-to verify A's signature on m

-compute v1=y**r.r**s mod p

-compute h(m) and v2=g**{h(m)} mod p

-the signature is accepted if and only if v1=v2

Applications of the DLP		Algorithms for Solving the DLP
	1971	Shank's Baby-Step Giant-Step
Diffie-Hellman - New Direction in Cryptography	1976	
	1978	Pollard's Rho and Lambda Methods; Pohlig-Hellman - GF(p)
	1979	Adleman - subexponential algorithm for GF(p)
	1980	
	1981	Herlestam-Johanesson- GF(2**p)
	1982	Hellman and Reyneri - GF(q)
	1983	ElGamal - subexponential algorithm for GF(p**2)
Odlyzko - DL in finite fields and their cryptographic significance; ElGamal - Public-key cryptosystem and signature scheme	1984	Coppersmith - subexponential algorithm for GF(2**n)
	1985	ElGamal - subexponential algorithm for GF(p**m)
	1986	Coppersmith,Odlyzko,Schroeppel subexponential algorithm for GF(p)
Chaum,Evertse and van de Graaf- protocols proving possession of DL	1987	
Agnew,Mullin and Vanstone - data exchange protocols	1988	Long and Wigderson - discrete log hides O(log n) bits
Schnorr - efficient identification and signatures for smart cards	1989	
Brickell and McCurley - interactive identification schemes	1990	Buchman and Dullman - Discrete log in class groups
Desmedt and Burmester-zero-knowledge schemes for DL based on smooth integers	1991	Schnorr - Compute discrete logs via diophantine approx.
Hardjano and Zheng - digital multi-signature	1992	Gordon and McCurley - massively parallel computation of discrete logs
	1993	Adleman and DeMarrais - subexponential algorithm for DL over all finite fields
Camenisch,Piveteau and Stadler -blind signature schemes based on DLP	1994	Maurer - equivalence between breaking DH and computing discrete logarithms
Horster,Michels and Petersen-Hidden signatures schemes based on DLP	1995	Weber - implementation of the GNFS to compute discrete logs mod p
Oorschot and Wiener-On Diffie-Hellman key agreement with short exponents	1996	Boneh and Venkatesan - Hardness of computing MSB of DL of secret keys in DH
	1997	Shoup - Lower bounds for DLP and related problems
	1998	Schnorr - security of almost all discrete log bits; Semaev-Discrete log in nonprime finite fields

TIME

Algorithms for Solving the DLP

In a group G of order n.

- algorithms which work in arbitrary groups :
  - **Exhaustive Search** :
    - Time : $O(n)$
    - Space : $O(1)$
    - Obs : deterministic algorithm
  - **Shank's Baby-Step Giant-Step** :
    - Time : $O(\sqrt{n})$
    - Space : $O(\sqrt{n})$
    - Obs : deterministic algorithm
  - **Pollard's Rho Method** :
    - Time : $O(\sqrt{n})$
    - Space : $O(1)$
    - Obs : probabilistic algorithm
  - **Pollard's Lambda Method**:
    - Time : $O(\sqrt{w})$
    - Space : $O(\log w)$
    - Obs :
      - probabilistic algorithm
      - w is the length of an interval in which the discrete logarithm lies.

- **Pohlig-Hellman** :
  - Time : $O(\sum_{i=1}^{r} e_i(\log n + \sqrt{p_i}))$ if the factorization of $n = \prod_{i=1}^{r} p_i^{e_i}$ is given
  - Space : $O(1)$
  - Obs: constant amount of space if Pollard's Rho method is used for calculating individual discrete logs.

- algorithms which work only in certain groups :
  - **Index Calculus** :
    - Time : $L_n(\frac{1}{3}, 1.923) = (1.923 + o(1)) \log^{\frac{1}{3}} n. \log\log^{\frac{2}{3}} n$ (using variant of Index Calculus - NFS)
    - Space : $L_n(\frac{1}{2}, c) = (c+o(1)) \log^{\frac{1}{2}} n. \log\log^{\frac{1}{2}} n$, c "small"
    - Obs : probabilistic algorithm

Time-Complexity of DL Algorithms

1971 ← Shank's Baby-Step Giant-Step - O(sqrt(n))

1976

1978 ← Pollard's Rho - O(sqrt(n))
Pollard's Lambda - O(sqrt(w))
Pohlig-Hellman

1979 ← Adleman - GF(p) - L_p(1/2,c)

1980

1981

1982 ← Hellman and Reyneri - GF(p**m) - subexponential

1983

1984 ← Coppersmith - GF(2**n)- L_n(1/3,c)
Blake,Fuji-Hara,Mullin and Vanstone-GF(2**n) - n=127

1985 ← ElGamal - GF(p**m) - subexponential
Odlyzko - GF(2**n) - subexponential

1986 ← Coppersmith,Odlyzko and Schroeppel- GF(p) - L_p(1/2,1)

1987 ← Pomerance
GF(p) - L_p(1/2, 1.414)
GF(2**m) - L_p(1/2, 1.414)

1988

1989

1990 ← LaMacchia and Odlyzko-GF(p)-Gaussian integer method
Index Calculus Variant

1991 ← Adleman and DeMarrais-GF(p**m) - L_p(1/2,c)

1992 ← Gordon, McCurley
GF(2**227),GF(2**313),GF(2**401)(5 days)
GF(2**503) (44 days),GF(2*593)

1993 ← Gordon - NFS -GF(p) -L_p(1/3,2.080)
Schirokauer - GF(p) - L_p(1/3, 1.923)

1994 ← Oorschot,Wiener-parallel Pollard's Rho
Adleman- Function Field Sieve -GF(p**m) - L_p**m(1/3,c)

1995 ← Weber - GNFS - GF(p) - log(p)=132

1996 ← Weber,Denny,Zayer-GF(p) -log(p)=248

1997

1998

TIME

Exhaustive Search - EXAMPLE

Suppose $G = Z_p^*$ with p a 200-bit prime number.
Using 2^{20} computers, each one running at 400 MHz, and assuming that one exponentiation ($g^i \mod p$) takes only one machine cycle (!), then $400.2^{20} \approx 2^{29}$ exponentiations can be made per second per computer. As there are around 2^{25} seconds in a year, it would take about $\frac{2^{200}}{2.2^{20+29+25}} \approx 2^{125}$ years to find the desired discrete logarithm, on average, by trial and error.

Baby-Step Giant-Step - DESCRIPTION

Input: a finite group $G =< g >$ of order n and $g, y \in G$.

Output: the discrete logarithm of y to the base g.

Procedure:

- calculate a table T of $(i, g^{i.\lceil\sqrt{n}\,\rceil})$ pairs, $1 \leq i \leq \lceil\sqrt{n}\,\rceil$, and sort them by the second component in the pair.
- compute $y.g^j$, starting at $j = 0$ ($y.g^j$ might be equal to g^{x+j} for some x and j).
- compare $y.g^j$ with the $g^{i.\lceil\sqrt{n}\,\rceil}$ entries in T. If a match is found then $g^{i.\lceil\sqrt{n}\,\rceil} = g^{x+j}$, and thus $x = i.\lceil\sqrt{n}\,\rceil - j$ is the desired discrete log. If there is no match, then try another value of j, with $j \leq \lceil\sqrt{n}\,\rceil$

Time-complexity: $O(\sqrt{n})$

Space-Complexity: $O(\sqrt{n})$

Baby-Step Giant-Step - EXAMPLE

Let $G = Z_{127}^*$. Then $g = 3$ is a generator of G (i.e has order $n = 126$). We look for the discrete logarithm of $y = 51$ to the base g.

First let's construct the table T of $(i, g^{i.\lceil\sqrt{127}\,\rceil})$ pairs, for $1 \leq i \leq \lceil\sqrt{127}\,\rceil = 12$:

i	1	2	3	4	5	6	7	8	9	10	11	12
$3^{i.\lceil\sqrt{127}\rceil} \mod 127$	73	122	16	**25**	47	2	19	117	32	50	94	4

Sorting T by the second entry:

i	6	12	3	7	4	9	5	10	1	11	8	2
$3^{i.\lceil\sqrt{127}\rceil} \mod 127$	2	4	16	19	**25**	32	47	50	73	94	117	122

Calculating $y.g^j$ for $j = 0, 1, 2, \ldots$ and comparing with T's entries:

j	0	1	2	3	4	5	6	7	8	9
$51.3^j \mod 127$	51	26	78	107	67	74	95	31	93	**25**

Therefore, $3^{4.12} = 3^{9+x}$, i.e the discrete logarithm of 51 to the base 3 in Z_{127}^* is $x = 48 - 9 = 39$.

Pollard's Rho (ρ) Method - DESCRIPTION

Input: a finite group $G =< g >$ of order n and $g, y \in G$.

Output: the discrete logarithm of y to the base g.

Procedure:

- break the set G into three approximately equal-sized sets S_1, S_2, S_3.
- define a sequence of elements over G: $x_0, x_1, x_2, \ldots$ by

$$x_0 = 1$$

$$x_{i+1} = f(x_i) = \begin{cases} y.x_i & \text{if } x_i \in S_1 \\ x_i^2 & \text{if } x_i \in S_2, \quad i \geq 0 \\ g.x_i & \text{if } x_i \in S_3 \end{cases}$$

- this sequence, in turn, defines two other integer sequences a_i, b_i such that $x_i = g^{a_i}.y^{b_i}$, for $i \geq 0$:

$$a_0 = b_0 = 0$$

$$a_{i+1} = \begin{cases} a_i & \text{if } x_i \in S_1 \\ 2.a_i & \text{if } x_i \in S_2, \quad i \geq 0 \\ a_i + 1 & \text{if } x_i \in S_3 \end{cases}$$

$$b_{i+1} = \begin{cases} b_i + 1 & \text{if } x_i \in S_1 \\ 2.b_i & \text{if } x_i \in S_2, \quad i \geq 0 \\ b_i & \text{if } x_i \in S_3 \end{cases}$$

- (Cycle-finding calculation) The next step is to find a pair (x_i, x_{2i}) with $x_i = x_{2i}$.

  In this case, $g^{a_i}.y^{b_i} = g^{a_{2i}}.y^{b_{2i}} \Rightarrow g^{a_{2i}-a_i} = y^{b_i - b_{2i}} \Rightarrow$ $\log_g g^{a_{2i}-a_i} = \log_g y^{b_i-b_{2i}} \Rightarrow \log_g y = \frac{a_{2i}-a_i}{b_i-b_{2i}} \mod n$.

  As long as $b_i \not\equiv b_{2i} \mod n$ the discrete logarithm can be calculated as above.

  In the rare case that a collision $x_i = x_{2i}$ is not found, or that $b_i \equiv b_{2i} \mod n$, the procedure can be repeated by selecting random $a_0, b_0 \in [1, n-1]$ and restarting with $x_0 = g^{a_0}.y^{b_0}$.

Time-complexity: $O(\sqrt{n})$

Space-Complexity: $O(1)$

Pollard's Rho (ρ) Method - EXAMPLE

Let $H = Z^*_{383}$. Then $g = 2$ is a generator of the subgroup G of Z^*_{383} of order $n = 191$. Suppose $y = 228$. Partitioning G into three sets according to the rule $x \in S_1$ if $x \equiv 1 \mod 3$; $x \in S_2$ if $x \equiv 0 \mod 3$; and $x \in S_3$ if $x \equiv 2 \mod 3$, and setting $x_0 = 0; a_0 = 0; b_0 = 0$, results :

i	x_i	a_i	b_i	x_{2i}	a_{2i}	b_{2i}
1	228	0	1	279	0	2
2	279	0	2	184	1	4
3	92	0	4	14	1	6
4	184	1	4	256	2	7
5	205	1	5	304	3	8
6	14	1	6	121	6	18
7	28	2	6	144	12	38
8	256	2	7	235	48	152
9	152	2	8	72	48	154
10	304	3	8	14	96	118
11	372	3	9	256	97	119
12	121	6	18	304	98	120
13	12	6	19	121	5	51
14	**144**	12	38	**144**	10	104

The calculations show that $x_{14} = x_{28} = 144$. Computing $r = (b_{14} - b_{28}) \mod 191 = 125$, and $r^{-1} = 125^{-1} \mod 191 = 136$ results in

$$x = \log_2 228 = r^{-1}.(a_{28} - a_{14}) \mod 191 = 110$$

Pollard's Lambda (λ) Method - DESCRIPTION (also affectionately called "Method for catching Kangaroos")

Input: a finite group $G =< g >$ of order n; $g, y \in G$ and a value w standing for the size of an interval in which the discrete logarithm lies (e.g. $A < \log_g y < B, w = B - A$).

Output: the discrete logarithm of y to the base g.

Procedure:

- compute two sequences T and W (called Kangaroo trails). The T sequence is $\{y'_0, y'_1, \ldots, y'_N\}$, where

$$y'_{i+1} = y'_i . g^{f(y'_i)}, i \geq 0$$

  where $f(.)$ is a "random" function, which takes values in a set R. At y'_N, T halts and "sets a trap" hoping to catch W in case it lands at this point during its trail $\{y_0, y_1, \ldots, y_M\}$, where

$$y_{i+1} = y_i . g^{f(y_i)}, i \geq 0$$

  Actually this will occur if W's trail hits any point $y'_i, 0 \leq i \leq N$. T's trail begins at $y'_0 = g^B \mod n$ and proceeds till y'_N. Note that $y'_{i+1} = y'_i . g^{f(y'_i)} \mod n \Rightarrow y'_N = y'_0 . g^{d_N} \mod n$, where $d_i = \sum_{j=0}^{i-1} f(y'_j)$ is the "distance" from y'_0 till y'_i.
  W's trail begins at $y_0 = y = g^x$. The search ends when $y_M = y'_N$ for some y_M in W's trail, at which point the discrete logarithm is calculated as :

$$y_M = y'_N \Rightarrow y_0.g^{\sum_{i=0}^{M-1} f(y_i)} = y'_0.g^{\sum_{j=0}^{N-1} f(y'_j)} \Rightarrow$$
$$y.g^{d_M} = g^B.g^{d_N} \Rightarrow g^{x+d_M} = g^{B+d_N} \Rightarrow$$
$$x = (B + d_N - d_M) \mod n$$

- if no collision $y_M = y'_N$ occurs (the probability of which can be controled) before d_M exceeds $B + d_N - A = w + d_N$ then the hunt is terminated with failure, because W has travelled beyond the trap. Subsequent iterations of the above sequences can be run as required.

Time-complexity: $O(\sqrt{w})$

Space-Complexity: $O(\log w)$

Pohlig-Hellman Algorithm - DESCRIPTION

Input: a finite group $G =< g >$ of order n, and $g, y \in G$.

Output: the discrete logarithm of y to the base g.

Procedure:

- let the factorization of n be $n = \prod_{i=1}^{r} p_i^{e_i}, e_i \geq 0$.
- for $i = 0$ to r compute the decomposition $x_i = l_0 + l_1.p_i + \ldots + l_{e_i-1}.p_i^{e_i-1}$ where $x_i = x \mod p_i^{e_i}$.
- set $\gamma = 1$; $l_{-1} = 0$ and $\overline{g} = g^{\frac{n}{p_i}}$.
- (compute l_j) for $j = 0$ to $e_i - 1$ do

  compute $\gamma = \gamma.g^{l_{j-1}.p_i^{j-1}}$ and $\overline{y} = (y.\gamma^{-1})^{\frac{n}{p_i^{j+1}}}$

  compute $l_j = \log_{\overline{g}} \overline{y}$ (e.g. using Pollard's rho method)

  set $x_i = l_0 + l_1.p_i + l_2.p_i^2 + \ldots + l_{e_i-1}.p_i^{e_i-1}$
- use the CRT to compute the discrete log x with $0 \leq x \leq n-1$ and $x \equiv x_i \mod p_i^{e_i}, 1 \leq i \leq r$

Time-complexity: $O(\sum_{i=1}^{r} e_i(\log n + \sqrt{p_i})))$, provided the factorization of n is given.

Space-Complexity: $O(1)$ if used in conjunction with Pollard's rho method, for example.

Pohlig-Hellman Algorithm - EXAMPLE

Let $G = Z_{251}^*$, $g = 71$ a generator of Z_{251}^* (i.e. of order $n = 250$). Let $y = 210$. The discrete logarithm is computed as follows:

- the prime factorization of n is $250 = 2.5^3$.
- (computing $x_1 \equiv x \mod 2$) compute $\overline{g} = g^{n/2} \mod 251 = 250$ and $\overline{y} = y^{n/2} \mod 251 = 250$. Then $x_1 = \log_{250} 250 = 1$.
- (computing $x_2 \equiv x \mod 5^3 = l_0 + l_1.5 + l_2.5^2$)
  - compute $\overline{g} = g^{n/5} \mod 251 = 20$.
  - compute $\gamma = 1$ and $\overline{y} = (g.\gamma^{-1})^{n/5} \mod 251 = 149$; l_0 is found as $l_0 = \log_{20} 149 = 2$.
  - compute $\gamma = \gamma.g^2 \mod 251 = 21$ and $\overline{y} = (g.\gamma^{-1})^{n/5} \mod 251 = 113$; l_1 is found as $l_1 = \log_{20} 113 = 4$.
  - compute $\gamma = \gamma.g^{4.5} \mod 251 = 115$ and $\overline{y} = (g.\gamma^{-1})^{(p-1)/5} \mod 251 = 149$; l_2 is found as $l_2 = \log_{20} 149 = 2$.
  - hence $x_2 = 2 + 4.5 + 2.5^2 = 72$.
- Solving the pair of congruences:

  $x \equiv 1 \mod 2$

  $x \equiv 72 \mod 125$

  gives the discrete logarithm

$$x = \log_{71} 210 = 197$$

Index Calculus - DESCRIPTION

Input: a finite group $G =< g >$ of order n and $g, y \in G$.

Output: the discrete logarithm of y to the base g.

Procedure:

- (Factor Base) find a subset $S = \{p_1, p_2, \ldots, p_t\}$ of G such that a significant fraction of all elements in G can be efficiently expressed as a product of elements from S.
- (Linear Relations)
  - select a random integer k, $0 \leq k \leq n-1$ and compute g^k.
  - try to write g^k as a product of elements from S:
    $g^k = \prod_{i=1}^{t} p_i^{c_i}$, $c_i > 0 \quad (*)$
  - if (*) is successful, then taking the discrete log of both sides of (*) results in the linear relation:
    $k = \sum_{i=1}^{t} c_i . \log_g p_i \mod n \quad (**)$
  - collect $t + c$ linear relations like (**), by choosing other random k's.
- (Discrete logs of elements of S) working modulo n, solve the linear system of $t + c$ equations in t unknowns and obtain the value of the discrete logarithms of the factor base: $\log_g p_i$, $1 \leq i \leq t$.

- (Actual computation of the discrete log) select a random r, $0 \leq r \leq n-1$ and compute $y.g^r$.
- try to write $y.g^r$ as a product of elements of S :
  $y.g^r = \prod_{j=1}^{t} p_j^{d_j}$, $d_j > 0$ (***)
- if $y.g^r$ is not representable as (***) then choose another r and retry the factorization (***). Otherwise, taking discrete logs of both sides of (***) results:
  $\log_g y = \sum_{i=1}^{t} d_i . \log_g p_i - r = x$

Time-complexity:

- $L_p(\frac{1}{3}, 1.923)$ for $G = Z_p^*$ using an NFS-variant of index calculus
- $L_{2^m}(\frac{1}{3}, c)$ for $G = GF(2^m)$ and $c < 1.587$

Space-Complexity:

- $L_p(\frac{1}{3}, 1.923)$ for $G = Z_p^*$
- $L_{2^m}(\frac{1}{3}, c)$ for $G = GF(2^m)$ and $c < 1.587$

Index Calculus - EXAMPLE

Let $G = Z^*_{229}$, $g = 6$ and $y = 13$. We wish to find $\log_6 13$.

- the factor base is chosen to be the first 5 primes $S = \{2, 3, 5, 7, 11\}$
- next linear relations are obtained from g^r for random values r:

  $6^{100} \mod 229 = 180 = 2^2.3^2.5$
  $6^{18} \mod 229 = 176 = 2^4.11$
  $6^{12} \mod 229 = 165 = 3.5.11$
  $6^{62} \mod 229 = 154 = 2.7.11$
  $6^{143} \mod 229 = 198 = 2.3^2.11$
  $6^{206} \mod 229 = 210 = 2.3.5.7$

- the following relations are obtained :

  $100 \equiv 2.\log_6 2 + 2.\log_6 3 + \log_6 5 \mod 228$
  $18 \equiv 4.\log_6 2 + \log_6 11 \mod 228$
  $12 \equiv \log_6 3 + \log_6 5 + \log_6 11 \mod 228$
  $62 \equiv \log_6 2 + \log_6 7 + \log_6 11 \mod 228$
  $143 \equiv \log_6 2 + 2.\log_6 3 + \log_6 11 \mod 228$
  $206 \equiv \log_6 2 + \log_6 3 + \log_6 5 + \log_6 7 \mod 228$

- solving the system of 6 equations in 5 unknowns (the individual logarithms $x_i = \log_6 p_i$) yields $\log_6 2 = 21$; $\log_6 3 = 208$; $\log_6 5 = 98$; $\log_6 7 = 107$; $\log_6 11 = 162$.
- choosing $k = 77$, and combining with y and g: $y.g^k = 13.6^{77} \mod 229 = 147 = 3.7^2$, so

  $\log_6 13 = (\log_6 3 + 2.\log_6 7 - 77) \mod 228 = 117$

Generalizations of the DLP

- Given a finite group G and $y, g \in G$, find the smallest positive integer x such that $y = g^x$, provided that such x exists. It is not required that G be cyclic or that g be a generator of G.
- DLP in subgroups of Z_p^*, p prime. Let G be the unique cyclic subgroup of Z_p^* of order q, $q \mid (p-1)$. Given $p, q, y, g \in G$, find x, $0 \leq x \leq q-1$ such that $g^x = y \mod p$. The index calculus does not seem to apply directly in G, but this method must be applied to Z_p^* itself in order to compute x for the subgroup G. Used in NIST's DSA (Digital Signature Algorithm).
- Group of points of an elliptic curve defined over a finite field

Conclusions

- Exhaustive search: infeasible for groups of large order.
- Baby-Step Giant-Step: inneficient for groups of large order.
- Pollard's Rho: inneficient for groups of large order.
- Pollard's Lambda: inneficient for groups of large order.
- Pohlig-Hellman: inneficient for groups of order n, where n has "large" prime factors (e.g. $n = 2.p' + 1$, p' prime).
- Index Calculus: (fastest algorithms):
  - DLP in $GF(p)$: Complexity $L_n(\frac{1}{2}, 1)$ (1986)
  - DLP in $GF(p^m)$: Complexity $L_{p^m}(\frac{1}{2}, c)$ (1991)
  - DLP in $GF(2^p)$, p prime : implemented successfully for $p < 593$ (Gordon and McCurley - 1992)
  - DLP in $GF(2^n)$: computed up to $n \approx 248$ (Weber - 1996)
    Complexity $L_{2^n}(\frac{1}{3}, c)$ (Coppersmith - 1984)

Conclusions (cont.)

Method	Complexity	Examples $p \approx 2^{200} \approx w$, $p = 2 * p' + 1$, p' prime, $c = 2$	Examples $p \approx 2^{593} \approx w$, $p = 2 * p' + 1$, p' prime, $c = 2$
Exhaustive Search	$O(p)$	$\approx 2^{200}$	$\approx 2^{593}$
Baby-Step Giant-Step	$O(\sqrt{p})$	$\approx 2^{100}$	$\approx 2^{296.5}$
Pollard's Rho	$O(\sqrt{p})$	$\approx 2^{100}$	$\approx 2^{296.5}$
Pollard's Lambda	$O(\sqrt{w})$	$\approx 2^{100}$	$\approx 2^{296.5}$
Pohlig Hellman	$O(\sum_{i=1}^{t} e_i(\log p + \sqrt{p_i}))$	$\approx 2^{99}$	$\approx 2^{296}$
Index Calculus	$L_p(1/3, c)$	$\approx 2^{69}$	$\approx 2^{110}$

References

[1] A.M.Odlyzko - Discrete logarithms and smooth polynomials - *Contemporary mathematics - Dec. 27, 1993* p. 1-10

[2] A.M.Odlyzko - Discrete logarithms in finite fields and their cryptographic significance - *Eurocrypt'84- LNCS - vol.209* - p. 1-87

[3] Bert den Boer - Diffie-Hellman is as strong as discrete log for certain primes - *Crypto'88 - LNCS - vol.403* - p.530-539

[4] Brian A. LaMacchia and A.M.Odlyzko - Computation of discrete logarithms in prime fields - (extended abstract) - *Crypto'90 - LNCS - vol.537* - p. 616-618

[5] C.P.Schnorr - Factoring integers and computing discrete logarithms via diophantine approximation - *Eurocrypt'91 - LNCS - vol. 547* - p. 281-293

[6] Damian Weber - An implementation of the general number field sieve to compute discrete logarithms mod p - *Eurocrypt'95 - LNCS - vol. 921* - p.95-105

[7] Damian Weber and Thomas Denny - The solution of McCurley's discrete log challenge - *Crypto'98 - LNCS - vol.1462* - p.458-471

[8] Dan Boneh, Ramarathiram Venkatesan - Hardness of computing the most significant bits of the secret keys in Diffie-Hellman and related schemes (extended abstract) - *Crypto'96 - LNCS - vol. 1109* - p. 129-142

[9] Daniel M. Gordon and Kevin S. McCurley - Massively parallel computation of discrete logarithms - *Crypto'92 - LNCS - vol.740* - p.312-323

[10] David Chaum, Jan-Hendrik Evertse and Jeroen van de Graaf - An improved protocol for demonstraring possession of discrete logarithms and some generalizations - *Eurocrypt'87 - LNCS - vol.304* - p.127-141

[11] Don Coppersmith, A.M.Odlyzko and Richard Schroeppel - Discrete logarithms in GF(p) - *Algorithmica - 1986* -p.1-15

[12] Eric Bach - Intractable problems in number theory - *Crypto'88 - LNCS - vol. 403* - p.77-93

[13] E.F.Brickell and J.H.Moore - Some remarks on the Herlestam-Johanneson algorithm for computing logarithms over GF(2^p) - *Crypto'82* - p.15-19

[14] E.F.Brickell and K.S.McCurley - An interactive identification scheme based on discrete logarithms and factoring - *Eurocrypt'90 - LNCS - vol.473* - p.63-71

[15] G.B.Agnew, R.C.Mullin and S.A.Vanstone - Data exchange protocol based on discrete exponentiation - *Eurocrypt'88 - LNCS- vol. 330* - p. 165-166

[16] I.F.Blake, R.C.Mullin and S.A.Vanstone - Computing logarithms in GF(2^n) - *Crypto'94 - LNCS - vol.196* - p.73-82

[17] Igor A.Semaev - An algorithm for evaluation of discrete logarithms in some nonprime finite fields - *Math. Comp. -vol.67 -no. 224 - Oct.98* -p.1679-1689

[18] Ingrid Biehl, Johannes Buchman and Christoph Thiel - Cryptographic protocols based on discrete logarithms in real quadratic orders - *Crypto'94 - LNCS - vol.839* - p.56-60

[19] Jan L. Camenisch, Jean-Marc Piveteau and Markus A. Stadler - Blind signatures based on the discrete logarithm problem - *Eurocrypt'94 - LNCS - vol. 950* - p. 428-432

[20] Johannes Buchman and Stephan Dullman - On the complexity and efficiency of a new key exchange system - *Eurocrypt'89 - LNCS - vol.434* - p.597-616

[21] Johannes Buchman and Stephan Dullman - On the computation of discrete logarithms in class groups (extended abstract) - *Crypto'90 - LNCS - vol. 537* - p. 134-139

[22] Kaisa Nyberg and Rainer A. Rueppel - Message recovery for signature schemes based on the discrete logarithm problem - *Eurocrypt'94 - LNCS - vol. 950* - p. 182-193

[23] Kouichi Sakurai and Hiroki Shizuya - Relationships among the computational powers of breaking discrete logarithm cryptosystems - *Eurocrypt'95 - LNCS - vol.921* - p.341-355

[24] Lein Harn and Shoubao Yang - Public-key cryptosystem based on the discrete logarithm problem - *Auscrypt'92 - LNCS - vol. 718* - p.469-476

[25] Leonard M.Adleman - A subexponential algorithm for the discrete logarithm problem with applications to cryptography (abstract) - *IEEE - 1979*

[26] Leonard M. Adleman and Jonathan DeMarrais - A subexponential algorithm for discrete logarithms over all finite fields - *Crypto'93 - LNCS - vol.773* - p.147-158

[27] Marc Girault - An identity-based identification scheme based on discrete logarithm modulo a composite number - *Eurocrypt'90 - LNCS - vol.473* - p.481-486

[28] Martin E.Hellman and Justin M. Reyneri - Fast computation of discrete logarithms in GF(q) - *Crypto'82* - p. 3-13

[29] Neal Koblitz - A family of Jacobian suitable for discrete logarithm cryptosystems - *Crypto'88 - LNCS - vol. 403* - p.94-99

[30] Patrick Horster and Hans-Joachim Knoblock - Discrete logarithm based protocols - *Eurocrypt'91 - LNCS - vol. 547* - p. 399-408

[31] Patrick Horster, Markus Michels and Holger Petersen - Meta message recovery and meta blind signature schemes based on the discrete logarithm

problem and their applications - *Asiacrypt'94 - LNCS - vol.917* - p.224-237

[32] Paul C. van Oorschot - A comparison of practical public-key cryptosystems based on integer factorization and discrete logarithms (extended abstract) - *Crypto'90 - LNCS - vol. 537* - p. 576-581

[33] Paul C. van Oorschot and Michael J. Wiener - On Diffie-Hellman key agreement with short exponents - *Eurocrypt'96 - LNCS - vol.1070* - p.332-343

[34] Peter Smith and Christopher Skinner - A public-key cryptosystem amd a digital signature scheme based on the Lucas function analogue to discrete logarithms - *Asiacrypt'94 - LNCS - vol.917*- p.357-364

[35] Rafi Heiman - A note on discrete logarithms with special strucuture - *Eurocrypt'92 - LNCS - vol.658* - p.454-457

[36] Rene Peralta - Simultaneous security of bits in the discrete log - *Eurocrypt'85 - LNCS - vol. 219* - p. 62-72

[37] Sarvar Patel and Ganapathy S.Sundaram - An efficient discrete log pseudo random generator - *Crypto'98 - LNCS - vol.1462* - p.304-317

[38] Shimshon Berkovits and Joel E.Sachs - Analysis of the Herlestam and Johanesson discrete logarithm scheme in GF(2^n) for large n - *BIT - 25 - 1985*- p.420-424

[39] Stephen C.Pohlig and Martin E.Hellman - An improved algorithm for computing logarithms over GF(p) and its cryptographic significance - *IEEE Trans. on Inof. Theory - vol.IT-24-no.1 - Jan 1978*

[40] Taher ElGamal - A public-key cryptosystem and a signature scheme based on discrete logarithm - *Crypto'94 - LNCS - vol. 196* - p.10-18

[41] Taher ElGamal - A subexponential-time algorithm for computing discrete logarithms over GF(p^2) - *IEEE Trans. on Info. Theory - vol.IT-31 - no. 4 - Jul. 1985* - p.473-481

[42] Taher ElGamal - On computing logarithms over finite fields - *Crypto'95 - LNCS - vol.218* - p.396-402

[43] Tatsuaki Okamoto, Kouichi Sakurai and Hiroki Shizuya - How intractable is the discrete logarithm for a general finite group ? - *Eurocrypt'92 - LNCS -vol.658* - p.420-428

[44] Thomas Hardjano and Yuliang Zheng - A practical digital multisignature scheme based on discrete logarithms (extended abstract) - *Auscrypt'92 - LNCS - vol. 718* - p.122-132

[45] Tore Herlestam and Rolf Johanesson - On computing logarithms over GF(2^p) - *BIT - 21 - 1981* - p.326-334

[46] Ueli M. Maurer - Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms - *Crypto'94 - LNCS - vol. 839* - p.271-281

[47] Ueli M.Maurer and Stefan Wolf - Lower bounds on generic algorithms in groups - *Eurocrypt'98 - LNCS - vol.1403* - p.72-84

[48] Yacov Yacobi - Discrete logarithms with compressible exponents - *Crypto'90 - LNCS - vol.537* - p. 639-643

[49] Yvo Desmedt and A.M.Odlyzko - A chosen-text attack on the RSA cryptosystem and some discrete logarithm schemes - *Crypto'85 - LNCS - vol.218* - p.516-522

[50] Yvo Desmedt and Mike Burmester - An efficient zero- knowledge scheme for the discrete logarithm based on smooth numbers - *Asiacrypt'91 - LNCS - vol.739* - p.360-367