Example of Bivariate Interpolation

Suppose that $p = 13$, $m = 2$, $y_1 = 1$, $y_2 = 2$, $y_3 = 3$
$a_1(x) = 1 + x + x^2$, $a_2(x) = 7 + 4x^2$ and $a_3(x) = 2 + 9x$.

$$\begin{aligned}
\frac{(y-2)(y-3)}{(1-2)(1-3)} &= 7y^2 + 4y + 3 \\
\frac{(y-1)(y-3)}{(2-1)(2-3)} &= 12y^2 + 4y + 10 \\
\frac{(y-1)(y-2)}{(3-1)(3-2)} &= 7y^2 + 5y + 1
\end{aligned}$$

$$\begin{aligned}
A(x,y) &= (1 + x + x^2)(7y^2 + 4y + 3) + (7 + 4x^2)(12y^2 + 4y + 10) \\
&\quad +(2 + 9x)(7y^2 + 5y + 1) \bmod 13 \\
&= y^2 + 3y + 10 + 5xy^2 + 10xy + 12x + 3x^2y^2 + 7x^2y + 4x^2
\end{aligned}$$

Insecurity wrt $k+1$ Colluders

- a set of bad users $W_1, \ldots, W_{k+1}$ (collectively) know the polynomials
$$g_{W_i}(x) = f(x, r_{W_i}) \bmod p,$$
$1 \leq i \leq k+1$
- using the bivariate interpolation formula, they can compute $f(x, y)$
- then they can compute any key

Security wrt k Colluders

- a set of bad users $W_1, \ldots, W_k$ (collectively) know the polynomials
$$g_{W_i}(x) = f(x, r_{W_i}) \bmod p,$$
$1 \leq i \leq k$

- we show that this information is consistent with any possible value of the key

- let K be the real (unknown) key, and let $K^* \neq K$

- define a polynomial $f^*(x, y)$ as follows:
$$f^*(x,y) = f(x,y) + (K^* - K) \prod_{1 \leq i \leq k} \frac{(x - r_{W_i})(y - r_{W_i})}{(r_U - r_{W_i})(r_V - r_{W_i})}$$

Security wrt k Colluders (cont.)

- f^* is a symmetric polynomial (i.e., $f(x,y) = f(y,x)$)
- for $1 \le i \le k$, it holds that

$$f^*(x, r_{W_i}) = f(x, r_{W_i}) = g_{W_i}(x)$$

- further,

$$f^*(r_U, r_V) = f(r_U, r_V) + K^* - K = K^*$$

- For any possible value of the key, K^*, there is a symmetric polynomial f^* such that the key $K_{U,V} = K^*$ and such that the secret information held by the k bad users is unchanged

Subgroups of Cyclic Groups (review)

- suppose that $(G, \cdot)$ is a cyclic group of order n
- let α be a generator of G (i.e., $\mathsf{ord}(\alpha) = n$)
- suppose that m is a divisor of n
- there is a unique subgroup H of G having order m
- the subgroup H is cyclic, and $\alpha^{n/m}$ is a generator of H (i.e., $\mathsf{ord}(\alpha^{n/m}) = m$)
- H consists of all the elements of G that have order dividing m
- if m is prime, then all elements of H other than the identity have order m (and hence they are all generators of H)

The Diffie-Hellman KPS

- the *Diffie-Hellman KPS* is a public-key based scheme to distribute secret LL-keys
- suppose α is an element having prime order q in the group ${\mathbb{Z}_p}^*$, where p is prime, $p-1 \equiv 0 \pmod q$, $p \approx 2^{1024}$ and $q > 2^{160}$
- α, p and q are public domain parameters
- every user U has a private LL-key a_U (where $0 \leq a_U \leq q-1$) and a corresponding public key

$$b_U = \alpha^{a_U} \bmod p$$

- the users' public keys are signed by the *TA* and stored on certificates, as usual

The Diffie-Hellman KPS (cont.)

- the secret LL-key $K_{U,V}$ for two users U and V is defined as follows:
$$K_{U,V} = \alpha^{a_U a_V} \bmod p$$
- V computes
$$K_{U,V} = {b_U}^{a_V} \bmod p,$$
using the public key b_U from U's certificate, together with his own secret key a_V
- U computes
$$K_{U,V} = {b_V}^{a_U} \bmod p,$$
using the public key b_V from V's certificate, together with her own secret key a_U

Security of the Diffie-Hellman KPS

- a coalition of bad users is of no help to the adversary in determining the key belonging to some disjoint pair of users
- the adversary's attempt to compute a key $K_{U,V}$ is an instance of the *Computational Diffie-Hellman* problem:

Problem: *Computational Diffie-Hellman (CDH)*
Instance: A multiplicative group $(G, \cdot)$, an element $\alpha \in G$ having order n, and two elements $\beta, \gamma \in \langle \alpha \rangle$.
Question: Find $\delta \in \langle \alpha \rangle$ such that

$$\log_\alpha \delta \equiv \log_\alpha \beta \times \log_\alpha \gamma \pmod{n}.$$

(Equivalently, given $\beta = \alpha^b$ and $\gamma = \alpha^c$, where b and c are unknown, compute $\delta = \alpha^{bc}$.)

Computational Diffie-Hellman $\propto_T$ Discrete Logarithm

- the *Computational Diffie-Hellman* problem is no harder to solve than the *Discrete Logarithm* problem in the same subgroup $\langle \alpha \rangle$
- given an oracle for the *DLP*, it is easy to solve the *CDH* problem, as follows:
- given inputs α, β, γ for *CDH*,
  1. use the oracle to compute $b = \log_\alpha \beta$
  2. compute $\delta = \gamma^b$
- the *Computational Diffie-Hellman* problem is thought to be infeasible when $G = \mathbb{Z}_p$ where $p \approx 2^{1024}$ is prime, n is a divisor of $p - 1$, and n has *at least one prime divisor* q *with* $q > 2^{160}$

Partial Information about Diffie-Hellman Keys

- the adversary may be unable to compute a Diffie-Hellman key but he could still (possibly) determine some partial information about the key
- we desire *semantic security* of the keys, which means that an adversary can compute no partial information about them (in polynomial time, say)
- in other words, distinguishing Diffie-Hellman keys from random elements of the subgroup $\langle \alpha \rangle$ should be infeasible
- semantic security of Diffie-Hellman keys is equivalent to the infeasibility of the *Decision Diffie-Hellman* problem

The Decision Diffie-Hellman Problem

Problem: *Decision Diffie-Hellman (DDH)*
Instance: A multiplicative group $(G, \cdot)$, an element $\alpha \in G$ having order n, and three elements $\beta, \gamma, \delta \in \langle \alpha \rangle$.
Question: Is it the case that $\log_\alpha \delta \equiv \log_\alpha \beta \times \log_\alpha \gamma \pmod{n}$? (Equivalently, given α^b, α^c and α^d, where b, d and d are unknown, determine if $d \equiv bc \pmod{n}$.)

- It is easy to see that the *Decision Diffie-Hellman* problem is no harder to solve than the *Computational Diffie-Hellman* problem in the same subgroup $\langle \alpha \rangle$

Decision Diffie-Hellman $\propto_T$ Computational Diffie-Hellman

- given an oracle for *CDH*, it is easy to solve the *DDH* problem, as follows:
- given inputs $\alpha, \beta, \gamma, \delta$ for *DDH*,
  1. use the oracle to to find the value δ' such that
  $$\log_\alpha \delta' \equiv \log_\alpha \beta \times \log_\alpha \gamma \pmod{n}$$
  2. check to see if $\delta' = \delta$

CDH in Cyclic Subgroups of Composite Order

- for a fixed α of order n, a triple $(\beta, \gamma, \delta) \in \langle\alpha\rangle \times \langle\alpha\rangle \times \langle\alpha\rangle$ that is a yes-instance of *DDH* is called a *Diffie-Hellman triple*
- there are n^3 triples in $\langle\alpha\rangle \times \langle\alpha\rangle \times \langle\alpha\rangle$, of which n^2 are Diffie-Hellman triples
- suppose α is an element of order n, and suppose that q is a proper prime divisor of n
- if q is "small" (e.g., $q \approx 2^{40}$), then it is easy to solve *DDH* for most triples
- for any $\beta \in \langle\alpha\rangle$, the Pohlig-Hellman algorithm can be used to compute $\log_\alpha \beta \bmod q$ in time $O(\sqrt{q})$

Pohlig-Hellman Algorithm

- let $a = \log_\alpha \beta$ and let $a_0 = a \bmod q$
- then $a = a_0 + Kq$ for some integer K
- then we have the following:

$$\begin{aligned}\beta^{n/q} &= (\alpha^a)^{n/q} \\ &= (\alpha^{a_0+Kq})^{n/q} \\ &= \alpha^{a_0 n/q}\alpha^{Kn} \\ &= \alpha^{a_0 n/q}\end{aligned}$$

Pohlig-Hellman Algorithm (cont.)

- since $\beta^{n/q} = \alpha^{a_0 n/q}$, where $0 \leq a_0 \leq q-1$, it is simple matter to determine a_0 by exhaustive search
- we begin by computing $\beta^{n/q}$ and $\gamma = \alpha^{n/q}$
- then we compute γ^i, $i = 0, \ldots, q-1$, by repeated multiplication by γ
- when we discover that
$$\gamma^i = \beta^{n/q}$$
for some i, where $0 \leq i \leq q-1$, we know that $a_0 = i$
- this gives a $O(q)$ algorithm, but a modification can reduce the complexity to $O(\sqrt{q})$

Solving DDH when q is small

- suppose we use the Pohlig-Hellman algorithm to compute

$$\begin{aligned} b_q &= \log_\alpha \beta \bmod q, \\ c_q &= \log_\alpha \gamma \bmod q, \quad \text{and} \\ d_q &= \log_\alpha \delta \bmod q \end{aligned}$$

- if (β, γ, δ) is a Diffie-Hellman triple, then

$$\log_\alpha \delta \equiv \log_\alpha \beta \log_\alpha \gamma \pmod{n},$$

and hence

$$d_q \equiv b_q c_q \pmod{q}$$

- therefore, if $d_q \not\equiv b_q c_q \pmod{q}$, then (β, γ, δ) is not a Diffie-Hellman triple
- hence, we can efficiently solve *DDH* for a $(q-1)/q$ fraction of the possible triples

Secruity of DDH

- the *Decision Diffie-Hellman* problem is thought to be infeasible when $G = \mathbb{Z}_p$ where $p \approx 2^{1024}$ is prime, n is a divisor of $p-1$, and n has *no prime divisor* q *with* $q < 2^{160}$
- this is a stronger condition than the one that is conjectured for the securiy of the *Computational Diffie-Hellman* problem