Lecture 8

Douglas Wikström
KTH Stockholm
dog@csc.kth.se

April 7, 2015

Discrete Logarithms

Discrete Logarithm Assumption

Let G_{q_n} be a cyclic group of prime order q_n such that $\lfloor \log_2 q_n \rfloor = n$ for $n = 2, 3, 4, \ldots$, and denote the family $\{G_{q_n}\}_{n\in\mathbb{N}}$ by G.

Definition. The **Discrete Logarithm (DL) Assumption** in G states that if generators g_n and y_n of G_{q_n} are randomly chosen, then for every polynomial time algorithm A

$$\Pr\left[A(g_n, y_n) = \log_{g_n} y_n\right]$$

is negligible.

Discrete Logarithm Assumption

Let G_{q_n} be a cyclic group of prime order q_n such that $\lfloor \log_2 q_n \rfloor = n$ for $n = 2, 3, 4, \ldots$, and denote the family $\{G_{q_n}\}_{n\in\mathbb{N}}$ by G.

Definition. The **Discrete Logarithm (DL) Assumption** in G states that if generators g and y of G are randomly chosen, then for every polynomial time algorithm A

$$\Pr\left[A(g, y) = \log_g y\right]$$

is negligible.

We usually remove the indices from our notation!

Diffie-Hellman Assumption

Definition. Let g be a generator of G. The **Diffie-Hellman (DH) Assumption** in G states that if $a, b \in \mathbb{Z}_q$ are randomly chosen, then for every polynomial time algorithm A

$$\Pr\left[A(g^a, g^b) = g^{ab}\right]$$

is negligible.

Decision Diffie-Hellman Assumption

Definition. Let g be a generator of G. The **Decision Diffie-Hellman (DDH) Assumption** in G states that if $a, b, c \in \mathbb{Z}_q$ are randomly chosen, then for every polynomial time algorithm A

$$\left|\Pr\left[A(g^a, g^b, g^{ab}) = 1\right] - \Pr\left[A(g^a, g^b, g^c) = 1\right]\right|$$

is negligible.

Relating DL Assumptions

- Computing discrete logarithms is at least as hard as computing a Diffie-Hellman element g^{ab} from g^a and g^b.
- Computing a Diffie-Hellman element g^{ab} from g^a and g^b is at least as hard as distinguishing a Diffie-Hellman triple (g^a, g^b, g^{ab}) from a random triple (g^a, g^b, g^c).
- In most groups where the DL assumption is conjectured, DH and DDH assumptions are conjectured as well.
- There exists special elliptic curves where DDH problem is easy, but DH assumption is conjectured!

Security of El Gamal

- Finding the secret key is equivalent to DL problem.
- Finding the plaintext from the ciphertext and the public key and is equivalent to DH problem.
- The semantic security of El Gamal is equivalent to DDH problem.

Brute Force and Shank's

Let G be a cyclic group of order q and g a generator. We wish to compute $\log_g y$.

- **Brute Force.** $O(q)$
- **Shanks.** Time and **Space** $O(\sqrt{q})$.
  1. Set $z = g^m$ (think of m as $m = \sqrt{q}$).
  2. Compute z^i for $0 \leq i \leq q/m$.
  3. Find $0 \leq j \leq m$ and $0 \leq i \leq q/m$ such that $yg^j = z^i$ and output $x = mi - j$.

Birthday Paradox

Lemma. Let $q_0, \ldots, q_k$ be randomly chosen in a set S. Then

1. the probability that $q_i = q_j$ for some $i \neq j$ is approximately $1 - e^{-\frac{k^2}{2s}}$, where $s = |S|$, and
2. with $k \approx \sqrt{-2s \ln(1-\delta)}$ we have a collision-probability of δ.

Proof.

$$\left(\frac{s-1}{s}\right)\left(\frac{s-2}{s}\right)\cdot\ldots\cdot\left(\frac{s-k}{s}\right) \approx \prod_{i=1}^{k} e^{-\frac{i}{s}} \approx e^{-\frac{k^2}{2s}} \quad .$$

Pollard-ρ (1/2)

Partition G into S_1, S_2, and S_3 "randomly".

- Generate "random" sequence $\alpha_0, \alpha_1, \alpha_2 \ldots$

$$\alpha_0 = g$$

$$\alpha_i = \begin{cases} \alpha_{i-1} g & \text{if } \alpha_{i-1} \in S_1 \\ \alpha_{i-1}^2 & \text{if } \alpha_{i-1} \in S_2 \\ \alpha_{i-1} y & \text{if } \alpha_{i-1} \in S_3 \end{cases}$$

Pollard-ρ (1/2)

Partition G into S_1, S_2, and S_3 "randomly".

- Generate "random" sequence $\alpha_0, \alpha_1, \alpha_2 \ldots$

$$\alpha_0 = g$$

$$\alpha_i = \begin{cases} \alpha_{i-1} g & \text{if } \alpha_{i-1} \in S_1 \\ \alpha_{i-1}^2 & \text{if } \alpha_{i-1} \in S_2 \\ \alpha_{i-1} y & \text{if } \alpha_{i-1} \in S_3 \end{cases}$$

- Each $\alpha_i = g^{a_i} y^{b_i}$, where $a_i, b_i \in \mathbb{Z}_q$ are known!

Pollard-ρ (1/2)

Partition G into S_1, S_2, and S_3 "randomly".

- Generate "random" sequence $\alpha_0, \alpha_1, \alpha_2 \ldots$

$$\alpha_0 = g$$

$$\alpha_i = \begin{cases} \alpha_{i-1} g & \text{if } \alpha_{i-1} \in S_1 \\ \alpha_{i-1}^2 & \text{if } \alpha_{i-1} \in S_2 \\ \alpha_{i-1} y & \text{if } \alpha_{i-1} \in S_3 \end{cases}$$

- Each $\alpha_i = g^{a_i} y^{b_i}$, where $a_i, b_i \in \mathbb{Z}_q$ are known!
- If $\alpha_i = \alpha_j$ and $(a_i, b_i) \neq (a_j, b_j)$ then $y = g^{(a_i - a_j)(b_j - b_i)^{-1}}$.

Pollard-ρ (2/2)

- If $\alpha_i = \alpha_j$, then $\alpha_{i+1} = \alpha_{j+1}$.
- The sequence $(a_0, b_0), (a_1, b_1), \ldots$ is "essentially random".
- The Birthday bound implies that the (heuristic) expected running time is $O(\sqrt{q})$.
- We use "double runners" to reduce memory.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.
  1. Choose $s_j \in \mathbb{Z}_q$ randomly and attempt to factor $g^{s_j} = \prod_i p_i^{e_{j,i}}$ as an **integer**.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.
  1. Choose $s_j \in \mathbb{Z}_q$ randomly and attempt to factor $g^{s_j} = \prod_i p_i^{e_{j,i}}$ as an **integer**.
  2. If g^{s_j} factored in $\mathcal{B}$ and $e_j = (e_{j,1}, \ldots, e_{j,B})$ is linearly independent of $e_1, \ldots, e_{j-1}$, then $j \leftarrow j + 1$.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.
  1. Choose $s_j \in \mathbb{Z}_q$ randomly and attempt to factor $g^{s_j} = \prod_i p_i^{e_{j,i}}$ as an **integer**.
  2. If g^{s_j} factored in $\mathcal{B}$ and $e_j = (e_{j,1}, \ldots, e_{j,B})$ is linearly independent of $e_1, \ldots, e_{j-1}$, then $j \leftarrow j + 1$.
  3. If $j < B$, then go to (1)

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.
- Repeat:
  1. Choose $s \in \mathbb{Z}_q$ randomly.
  2. Attempt to factor $yg^s = \prod_i p_i^{e_i}$ as an **integer**.
  3. If a factorization is found, then output $\left(\sum_i a_i e_i - s\right) \bmod q$.

Excercise: Why doesn't this work for any cyclic group?

Example Groups

- $\mathbb{Z}_n$ additively? **Bad for crypto!**

Example Groups

- $\mathbb{Z}_n$ additively? **Bad for crypto!**
- Large prime order subgroup of $\mathbb{Z}_p^*$ with p prime. In particular $p = 2q + 1$ with q prime.

Example Groups

- $\mathbb{Z}_n$ additively? **Bad for crypto!**
- Large prime order subgroup of $\mathbb{Z}_p^*$ with p prime. In particular $p = 2q + 1$ with q prime.
- Large prime order subgroup of $\mathrm{GF}_{p^k}^*$.

Example Groups

- $\mathbb{Z}_n$ additively? **Bad for crypto!**
- Large prime order subgroup of $\mathbb{Z}_p^*$ with p prime. In particular $p = 2q + 1$ with q prime.
- Large prime order subgroup of $\mathrm{GF}_{p^k}^*$.
- "Carefully chosen" elliptic curve group.

Elliptic Curves

Groups

- We have argued that discrete logarithm problems are hard in large subgroups of $\mathbb{Z}_p^*$ and $\mathbb{F}_q^*$.
- Based on discrete logarithm problems (DL, DH, DDH) we can construct public key cryptosystems, key exchange protocols, and signature schemes.
- An elliptic curve is another candidate of a group where discrete logarithm problems are hard.

Motivation For Studying Elliptic Curves

- What if it turns out that solving discrete logarithms in $\mathbb{Z}_p^*$ is easy? Elliptic curves give an **alternative**.
- The best known DL-algorithms in an elliptic curve group with prime order q are **generic algorithms**, i.e., they have running time $O(\sqrt{q})$
- Arguably we can use **shorter keys**. This is very important in some practical applications.

Definition

Definition. A plane cubic curve E (on Weierstrass form) over a field $\mathbb{F}$ is given by a polynomial

$$y^2 = x^3 + ax + b$$

with $a, b \in \mathbb{F}$. The set of points (x, y) that satisfy this equation over $\mathbb{F}$ is written $E(\mathbb{F})$.

Definition

Definition. A plane cubic curve E (on Weierstrass form) over a field $\mathbb{F}$ is given by a polynomial

$$y^2 = x^3 + ax + b$$

with $a, b \in \mathbb{F}$. The set of points (x, y) that satisfy this equation over $\mathbb{F}$ is written $E(\mathbb{F})$.

Every plane cubic curve over a field of characteristic $\neq 2, 3$ can be written on the above form without changing any properties we care about.

Alternative Notation

We also write

$$g(x, y) = x^3 + ax + b - y^2 \quad \text{or}$$
$$y^2 = f(x)$$

where $f(x) = x^3 + ax + b$.

Singular Points

Definition. A point $(u, v) \in E(\mathbb{E})$, with $\mathbb{E}$ an extension field of $\mathbb{F}$, is **singular** if

$$\frac{\partial g(x,y)}{\partial x}(u,v) = \frac{\partial g(x,y)}{\partial y}(u,v) = 0 \ .$$

Definition. A plane cubic curve is **smooth** if $E(\overline{\mathbb{F}})$ contains no singular points[1].

[1] $\overline{\mathbb{F}}$ is the algebraic closure of $\mathbb{F}$.

What Does This Mean?

Note that

$$\frac{\partial g(x,y)}{\partial x}(x,y) = f'(x) = 3x^2 + a \quad \text{and}$$
$$\frac{\partial g(x,y)}{\partial y}(x,y) = -2y \ .$$

Thus, any singular point $(u,v) \in E(\mathbb{F})$ must have:

- $v = 0$,
- $f(u) = 0$, and $f'(u) = 0$.

Then $f(x) = (x-u)h(x)$ and $f'(x) = h(x) + (x-u)h'(x)$, so (u,v) is singular if $v = 0$ and u is a double-root of f.

Discriminant

In general a “discriminant” can be used to check if a polynomial has a double root.

Definition. The discriminant $\Delta(E)$ of a plane curve $y^2 = x^3 + ax + b$ is given by $-4a^3 - 27b^2$.

Lemma. The polynomial $f(x)$ does not have a double root iff $\Delta(E) \neq 0$, in which case the curve is called **smooth**.

Line Defined By Two Points On Curve

Let $l(x)$ be a line that intersects the curve in (u_1, v_1) and (u_2, v_2). Then

$$l(x) = k(x - u_1) + v_1$$

where

$$k = \begin{cases} \frac{v_2 - v_1}{u_2 - u_1} & \text{if } (u_1, v_1) \neq (u_2, v_2) \\ \frac{3u_1^2 + a}{2v_1} & \text{otherwise} \end{cases}$$

Line Defined By Two Points On Curve

Let $l(x)$ be a line that intersects the curve in (u_1, v_1) and (u_2, v_2). Then

$$l(x) = k(x - u_1) + v_1$$

where

$$k = \begin{cases} \frac{v_2 - v_1}{u_2 - u_1} & \text{if } (u_1, v_1) \neq (u_2, v_2) \\ \frac{3u_1^2 + a}{2v_1} & \text{otherwise} \end{cases}$$

We are cheating a little here in that we assume that we don't have $u_1 = u_2$ and $v_1 \neq v_2$ or $v_1 = v_2 = 0$. In both such cases we get a line parallel with $x = 0$ that we deal with in a special way.

Finding the Third Point

- The intersection points between $l(x)$ and the curve are given by the zeros of

$$t(x) = g(l(x), x) = f(x) - l(x)^2$$

which is a cubic polynomial with known roots u_1 and u_2.

Finding the Third Point

- The intersection points between $l(x)$ and the curve are given by the zeros of

$$t(x) = g(l(x), x) = f(x) - l(x)^2$$

which is a cubic polynomial with known roots u_1 and u_2.

- To find the third intersection point (u_3, v_3) we note that

$$t(x) = (x - u_1)(x - u_2)(x - u_3) = x^3 - (u_1 + u_2 + u_3)x^2 + r(x)$$

where $r(x)$ is linear. Thus, we can find u_3 from t's coefficients!

From Intersection Points To Group Law

- Given any two points A and B the on the curve that defines a line, we can find a third intersection point C with the curve (even if $A = B$).

From Intersection Points To Group Law

- Given any two points A and B the on the curve that defines a line, we can find a third intersection point C with the curve (even if $A = B$).
- The only exception is if our line $l(x)$ is parallel with the y-axis.

From Intersection Points To Group Law

- Given any two points A and B the on the curve that defines a line, we can find a third intersection point C with the curve (even if $A = B$).
- The only exception is if our line $l(x)$ is parallel with the y-axis.
- To “fix” this exception we add a point at infinity O, roughly at $(0, \infty)$ (the projective plane). Intuition: the sides of a long straight road seem to intersect infinitely far away.

From Intersection Points To Group Law

- We define the sum of A and B by $(x, -y)$, where (x, y) is the third intersection point of the line defined by A and B with the curve.
- We define the inverse of (x, y) by $(x, -y)$.
- The main technical difficulty in proving that this gives a group is to prove the associative law. This can be done with Bezout's theorem (not the one covered in class), or by (tedious) elementary algebraic manipulation.

Elliptic Curves

- There are many elliptic curves with special properties.
- There are many ways to represent the same curve and to implement curves as well as representing and implementing the underlying field.
- More requirements than smoothness must be satisfied for a curve to be suitable for cryptographic use.

Elliptic Curves

- There are many elliptic curves with special properties.
- There are many ways to represent the same curve and to implement curves as well as representing and implementing the underlying field.
- More requirements than smoothness must be satisfied for a curve to be suitable for cryptographic use.
- Fortunately, there are **standardized curves**.

  (I would need a **very strong** reason not to use these curves and I would be **extremely careful**, consulting researchers specializing in elliptic curve cryptography.)