Crypto math II

Alin Tomescu
alinush@mit.edu

May 19, 2015

Abstract

A quick overview on group theory from Ron Rivest's 6.857 course in Spring 2015.

1 Overview

- Group theory review
- Diffie-Hellman (DH) key exchane
- Five crypto groups:
  - $\mathbb{Z}_p^*$
  - $\mathbb{Q}_p$
  - $\mathbb{Z}_n^*$
  - $\mathbb{Q}_n$
  - elliptic curves

2 Group theory review

Here, we are talking about multiplicative groups (where the operation between group elements is something *resembling* multiplication)

Definition: $(\mathbb{G}, \cdot)$ is a *finite abelian group* of size t if:

- $\exists$ identity 1 such that $\forall a \in \mathbb{G}, a \cdot 1 = 1 \cdot a = a$
- $\forall a \in \mathbb{G}, \exists b \in \mathbb{G}$ such that $a \cdot b = 1$
- $\forall a, b, c \in \mathbb{G}, a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- $\forall a, b \in \mathbb{G}, a \cdot b = b \cdot a$

2.1 Order and generators

Definition: The *order* of a in $\mathbb{G}$ is denoted by $order(a)$ and is equal to the least u such that $a^u = 1$

Lagrange's Theorem: In a finite abelian group of size t, for all $a \in \mathbb{G}$, $order(a) \mid t$

Theorem: In a finite abelian group of size t, $\forall a \in \mathbb{G}, a^t = 1$

Example: $a^{(p-1)} = 1, \forall a \in \mathbb{Z}_p^*$ because $|\mathbb{Z}_p^*| = 1$

Definition: $\langle a \rangle = \{a^i : i \geqslant 0\}$ = subgroup generated by a.

Definition: If $\langle a \rangle = \mathbb{G}$ then $\mathbb{G}$ is *cyclic* and a is a *generator* of $\mathbb{G}$.

Note: $|\langle a \rangle| = order(a)$

Exercise: In a finite abelian group $\mathbb{G}$ of order t, where t is prime, we have: $\forall a \in \mathbb{G}$, if $a \neq 1 \Rightarrow a$ is a generator of $\mathbb{G}$.

Solution: We know that the size of any subgroup of $\mathbb{G}$ must divide t. Since t is prime, any subgroup can either have size 1 or t. Thus, only trivial subgroups can exist: the subgroup made up of the identity element ($\{1\}$) and $\mathbb{G}$ itself. Since $a \neq 1$, any subgroup generated by a cannot be equal to $\{1\}$ because it will have to contain a itself which is different than 1. Thus, if a generates any subgroup, it has to generate $\mathbb{G}$ itself. How do we know that a generates any subgroup at all then? We know $a \in G \Rightarrow a^u \in G, \forall u$ and, informally, we know that there cannot be a $u, 1 < u < t$ such that $a^u = 1$ because that would create a subgroup of $\mathbb{G}$ of size u, which would imply $u \mid t$, which would be false since t is prime.

Theorem: $\mathbb{Z}_p^*$ is always cyclic (i.e. there exists a generator within $\mathbb{Z}_p^*$)

2.2 Discrete logs

Theorem: If $\mathbb{G}$ is a cyclic group of order t and generator g then the relation $x \leftrightarrow g^x$ is one-to-one between $[0, 1, \ldots, t-1]$ and $\mathbb{G}$.

$$x \mapsto g^x : \text{exponentiation, "powering-up"}$$
$$g^x \mapsto x : \text{discrete logarithm (DL)}$$

Computing *discrete logarithms* (the DL problem) is commonly assumed to be hard/infeasible for well-chosen groups $\mathbb{G}$ (e.g. $\mathbb{Z}_p^*$ for p a large randomly chosen prime).

In practice, we need to be able to translate *bits of data* or *messages* from a message space M to group elements of $\mathbb{G}$. We need an one-to-one (injective) function $f : M \to \mathbb{G}$ such that $f(m) \in \mathbb{G}$ can be chosen to represent message $m \in M$.

TODO: Does f need to be onto (surjective) as well? Are there cases where we need to reverse f?

Example: If we have $\mathbb{Z}_p^*$ with $p > 2^k$, then we can represent any k-bit message m as a number $x \in [0, 2^k)$ because if $x \in [0, 2^k)$, then $x \in \mathbb{Z}_p^*$.

Note: For some groups, finding an easily-computable, space-efficient f may be a little hard.

2.3 API for a group

Typically, any library that implements a group should provide the following calls:

Operation	API call	Comments
creation	$\mathbb{G} \leftarrow createGroup(\dots)$	
identity	$\mathbb{G}.identity()$	
random element	$x \leftarrow \mathbb{G}.random()$	
product	$x \cdot y$	or $+$
inverse	x^{-1}	or $-x$
power	$x^k, x \in \mathbb{G}, k \in \mathbb{Z}$	or $k \cdot x$
size	$\mathbb{G}.order()$	$\lvert G \rvert$, not always implemented
list	$\mathbb{G}.elements()$	not always implemented
represent	$x \leftarrow G.rep(m)$	
unrepresent	$m \leftarrow G.unrep(x)$	
generator	$\mathbb{G}.generator()$	
discrete log	$x \leftarrow \mathbb{G}.discreteLog(g, y)$	s.t $g^x = y$, not always *efficiently* possible

3 Diffie-Hellman key exchange (1976)

How can we establish a shared secret in the presence of a passive eavesdropper Eve?

Let $\mathbb{G}$ be a cyclic group with generator g. $\mathbb{G}$ and g are fixed and public.

Alice and Bob can agree on a shared secret key k as follows:

1. Alice chooses secret x randomly from $[0, \dots, |\mathbb{G}| - 1]$. Note that $x \notin \mathbb{G}$.
2. Alice computes g^x as her *public key*. Note that $g^x \in \mathbb{G}$ and Alice is the only one who knows x, the *discrete log* of g^x.
3. Bob, similarly picks a y and computes g^y.
4. Alice and Bob exchange g^y and g^x. Eve sees them.
5. Assuming discrete logs are hard to compute, Eve cannot learn neither x nor y because she will have a hard time computing the discrete log of g^x or g^y.
6. Alice computes $k = (g^y)^x = g^{xy}$
7. Bob computes (the same) $k = (g^x)^y = g^{xy}$
8. Alice and Bob have agreed on a shared key k
9. Can Eve compute g^{xy} from g^x and g^y? We assume she can't and refer to this assumption as the *Computational Diffie-Hellman* (CDH) assumption.

Theorem: CDH is hard $\Rightarrow$ Diffie-Hellman key exchange is secure (i.e. Eve does not learn k)

Note: Can use k to encrypt and/or MAC messages.

Note: If not using an authenticated encryption mode like EAX, derive separate keys for encryption and authentication $k_{enc} = PRF(k, enc)$ and $k_{mac} = PRF(k, mac)$.

Note: g^x and g^y are assumed to be the right public keys for Alice and Bob. What if Eve is active and changes them on their way to Alice and Bob?

If Eve is *active*, she can perform a *man-in-the-middle attack*:

1. Alice sends g^x to Bob, but Eve replaces it with g^e, for which she knows e. Eve records g^x.
2. Bob sends g^y to Bob, but Eve replaces it with g^v, for which she knows v. Eve records g^y.
3. Alice got g^v from Eve, so she will compute shared key $k_1 = g^{xv}$
4. Bob got g^e from Eve, so she will compute shared key $k_2 = g^{ye}$
5. Alice and Bob think they are talking to each other, but they agreed to different keys.
6. Eve can compute $k_1 = g^{xv}$ herself: she knows v
7. Eve can compute $k_2 = g^{ye}$ herself: she knows e
8. When Alice sends a message to Bob, encrypted and/or MACd with k_1, Eve can decrypt and tamper with the message and then reencrypt and MAC it under k_2 for Bob.
9. Eve can do the same for Bob's messages to Alice.

To fix this problem, we need to prevent Eve from swapping Alice and Bob's public keys on the wire. One solution is to have a *certification authority* (CA) digitally sign g^x and g^y so that Eve cannot replace them.

Note: Still not perfect. What if Eve colludes with the CA?

Note: What if Eve has friends with public keys signed by the CA. Those friends can maybe give Eve their private keys and Eve could still pull the attack $\Rightarrow$ the digital signature has to *cryptographically bind* the user's identity (Alice) to her public keys g^x. This way, if Eve replaces Alice's public key with her friend's Diana public key, Bob will detect this when he verifies the signature on the public key: the signature will not verify against Alice's name.

4 The five groups

4.1 $\mathbb{Z}_p^*$

Definition: $\mathbb{Z}_p^* = \{a : 1 \leqslant a < p\}$, where p is prime

$\mathbb{Z}_p^*$ is always cyclic (i.e. has a generator). There are non-constructive proofs for this.

If $p = 2q + 1$ and q is prime, then p is a *safe prime* and half of $\mathbb{Z}_p^*$ elements are generators and the other half are squares $\mathbb{Q}_p$.

TODO: Proof?

4.2 $\mathbb{Q}_p$, quadratic residues (squares) mod prime p

Definition: $\mathbb{Q}_p = \{a^2 : 1 \leqslant a < p\} \subsetneq \mathbb{Z}_p^*$

TODO: Is $a < p$ or $a^2 < p$?

Theorem: $|\mathbb{Q}_p| = \frac{1}{2}|\mathbb{Z}_p^*| = \frac{(p-1)}{2}$

Theorem: $\mathbb{Q}_p$ is cyclic: If $\langle g \rangle = \mathbb{Z}_p^*$, then $\langle g^2 \rangle = \mathbb{Q}_p$

Thus, $\mathbb{Q}_p = \{g^{2i} : 0 \leqslant i < \frac{p-1}{2}\}$, if $\langle g \rangle = \mathbb{Z}_p^*$

If $p = 2q + 1$, then $|\mathbb{Q}_p| = \frac{(p-1)}{2} = q$ and *any element* of $\mathbb{Q}_p$ (other than 1) generates $\mathbb{Q}_p$. To find a generator, take the square of any element $a \in \mathbb{Z}_p^* - \{1, p-1\}$

TODO: Proof for why $\mathbb{Z}_p^*$ is split that way? Is $p - 1$ the only generator that would generate a subgroup of size 2? The identity (i.e. 1) would generate the subgroup of size 1, and apparently all squares generate $\mathbb{Q}_p$ of size q, which means the rest either generate $\mathbb{Z}_p^*$ or the subgroup of size 2.

4.3 $\mathbb{Z}_n^*$

Definition: $\mathbb{Z}_n^* = \{a : \gcd(a, n) = 1, \text{where } 1 \leqslant a < n\}$

Definition: $|\mathbb{Z}_n^*| = \phi(n)$, the *totient* function.

If $n = pq$ where p, q are distinct odd primes, then $\mathbb{Z}_n^*$ is *not* cyclic.

...but the *Chinese Remainder Theorem* says there exists an isomorphism from $\mathbb{Z}_n^*$ to $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$.

4.4 $\mathbb{Q}_n$, quadratic residues (squares) mod (non-prime) n

Definition: $\mathbb{Q}_n = \{a^2 : 1 \leqslant a < n, \text{where } \gcd(a, n) = 1\}$

TODO: Is $a < p$ or $a^2 < p$?

TODO: Is $\gcd(a, n) = 1$ or $\gcd(a^2, n) = 1$?

Theorem: If $n = pq$ where $p = 2r + 1$ and $q = 2s + 1$, then $|\mathbb{Q}_n| = rs$ and $\mathbb{Q}_n$ is cyclic.

4.5 TODO: Elliptic curves