Today - Special Topic: Cryptography

- Commitments
- Zero-Knowledge Proofs

Some problems are hard...

- Consider the group $\mathbb{G} = \mathbb{Z}_p^*$ for some prime number p
- Let g be a non-identity element in $\mathbb{G}$
- Example: $p = 17$, $\mathbb{G} = \mathbb{Z}_{17}^* = \{1, 2 \cdots 16\}$
- Say $g = 3$, then what is $3^0 \mod 17 = 1$, $3^1 \mod 17 = 3$, $3^2 \mod 17 = 9$, $3^3 \mod 17 = 10$, $3^4 \mod 17 = 13 \ldots$, $3^{16} \mod 17 = 1$. Fermat's Little Theorem
- Given $x \in \mathbb{Z}$ can you compute $g^x \mod p$? Efficiently?
- What about the other way around? Given g, X, p can we compute x such that $X = g^x \mod p$?
- Efficiently? Well, it depends on what x was?
- Discrete-Log Problem: Sample (uniform) $x \xleftarrow{\$} \{1, \cdots p-1\}$ and give you g, X, p where $X = g^x \mod p$. Now can you find x?
- Best Algorithm: $e^{(3^{2/3} - o(1))(\log p)^{\frac{1}{3}} (\log\log p)^{\frac{2}{3}}}$

How large can primes be?

- The number of prime numbers is infinite.
- As of January 2017, the largest known prime number is $2^{74,207,281} - 1$, a number with $22,338,618$ digits. It was found in 2016 by the Great Internet Mersenne Prime Search (GIMPS).
- Using large enough primes primes the discrete log problem is believed to be hard!

Commitment Schemes

- A protocol between a committer (C) and a receiver (R)
- C's input: a bit $b \in \{0,1\}$ and R has no input
- **Commitment Phase**: $\langle C(b; s_C) \leftrightarrow R(s_R) \rangle$

  Opening Phase: C sends b, s_C to R who outputs 0 or 1.

  - **Correctness**: If C and R are honest then R always outputs 1
  - **Hiding**: At the end of the commitment phase, R doesn't learn anything about b.
  - **Binding**: C can not find $(0, s_0)$ and $(1, s_1)$ such that R outputs 1 on both.

Commitment Protocol

$$
\begin{array}{ccc}
\textit{Commiter}(b; s_C) & & \textit{Receiver}(s_R) \\
 & & x \leftarrow \{0, \cdots p-1\} \\
 & & h := g^x \mod p \\
 & \xleftarrow{\hspace{4em} h \hspace{4em}} & \\
 & \xrightarrow{\hspace{2em} Y = g^b h^{s_C} \hspace{2em}} & \\
 & & \textit{Store } Y \\
\textit{Opening Phase} & & \\
 & \xrightarrow{\hspace{2em} b, s_C \hspace{2em}} & \\
 & & \text{Output 1 if } g^b h^{s_C} \stackrel{?}{=} Y \\
 & & \text{Else output 0}
\end{array}
$$

Is it hiding?

- Y contains no information about b.
- If $g^b h^s = Y$ then $g^{1-b} h^{s'} = Y$ where $s' = \frac{2b-1}{x} + s \mod p-1$.[1]

[1] For this class, we ignore that x^{-1} may sometimes not exist.

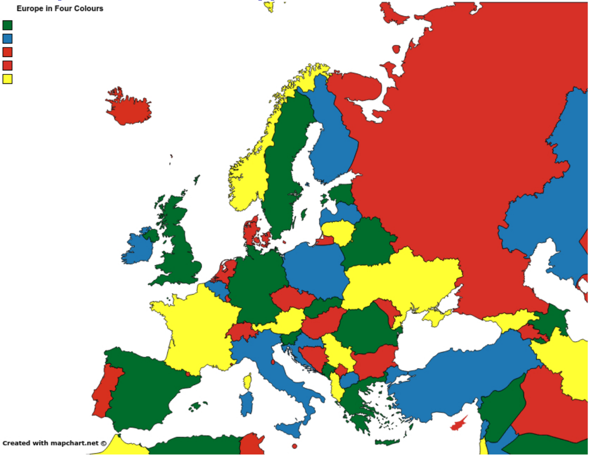
Is it binding?

- It is only computationally binding!
- If at the end of the protocol C can come up with $(0, s_0)$ and $(1, s_1)$ such that R outputs 1 on both choices then we can use this "procedure" to solve the discrete-log problem.
- Given (g, X, p) we are trying to find $dlog_g\ X$.We set $h = X$ on behalf of R.Now given $(0, s_0)$ and $(1, s_1)$ (and because R outputs 1 on both) we have that $x \cdot s_0 = 1 + x \cdot s_1$.Therefore, $x = \frac{1}{s_0 - s_1}$ mod $p - 1$

How would you prove that a NP problem is true?

- A *NP problem* I is true if there exists a solution S such that $\mathscr{C}(I,S) = \textit{true}$, where $\mathscr{C}$ is the checking algorithms.
- You can send the solution S to your friend.
- However, this leaks the solution to your friend.

Graph 3-coloring



- Can you color a map in 3 colors?
- How can you prove to a friend that there exists a 3-coloring without disclosing the coloring itself?
- This problem is NP-complete.

Zero-Knowledge Proofs

- We have two players: a prover ($\mathscr{P}$) and a verifier ($\mathscr{V}$)
- $\mathscr{P}$ and $\mathscr{V}$ get as input a graph/map $G = (V, E)$
- $\mathscr{P}$ also gets as input a coloring function $c : V \to \{R, B, G\}$.
- A protocol $\langle \mathscr{P}, \mathscr{V} \rangle$ where at the end $\mathscr{V}$ outputs 0 or 1.
  - Correctness: Execution with honest $\mathscr{P}, \mathscr{V}$ always leads $\mathscr{V}$ to output 1.
  - Soundness: For any cheating $\mathscr{P}^*$ and G that is not 3-colorable $\mathscr{V}$ outputs 0 with probability greater that $1 - 2^{-\lambda}$.
  - Zero-Knowledge: No cheating $\mathscr{V}^*$ learns anything about P's coloring function c.

Zero-Knowledge Protocol

$$\mathscr{P}(G, c; r)$$

π be a random function

$$\{R, B, G\} \to \{R, B, G\}$$

$$\mathscr{V}(G, s)$$

$$\xrightarrow{\forall v \in V, c_v = com(\pi(c(v)))}$$

$$e \xleftarrow{\$} E$$

$$\xleftarrow{e = (u, v)}$$

$$\xrightarrow{\text{open } c_u, c_v}$$

Output 1
if diff
Else 0

Correctness and Soundness

- If $\mathscr{P}, \mathscr{V}$ are honest then does V always accept?
- What is G doesn't have any 3-colorings? $\mathscr{V}$ catches the prover with probability $\frac{1}{|E|}$.
- How do we reduce probability of not catching to $2^{-\lambda}$? Repeat it $|E| \cdot \lambda$ times.
- Must use fresh randomness (namely π) in each.

Zero-Knowledge

- What does a cheating verifier $\mathscr{V}^*$ learn in one execution?
- Nothing! :)

- CS194 on Cryptography: Next Semester