Lecture 27: DDH Assumption, Key Agreement, and ElGamal Encryption
The objective of this lecture is to build key agreement and public-key encryption protocols from the Decisional Diffie-Hellman (DDH) assumption.

Moreover, understand the relationship between the DDH assumption and other computational hardness assumptions like the discrete log assumption and Computational Diffie-Hellman (CDH) assumption.
Consider a group \((G, \times)\) with generator \(g\) and order \(n\); i.e., \(g^n = e\), the identity and \(\{g^1, g^2, \ldots, g^n = e\} = G\).

The Decisional Diffie-Hellman (DDH) assumption states that it is computationally infeasible to have a non-trivial advantage in predicting whether the given sample \((\alpha, \beta, \gamma) \in G^3\) was sampled from the distribution \((g^a, g^b, g^r)\), where \(a, b, r \in \mathbb{R}\{1, 2, \ldots, n\}\), or \((g^a, g^b, g^{ab})\), where \(a, b \in \mathbb{R}\{1, 2, \ldots, n\}\).

Intuitively, given \((g^a, g^b)\), the element \(g^{ab}\) is computationally indistinguishable from the random \(g^r\).
Diffie-Hellman Key Agreement

1. Alice samples $a \in \mathbb{R} \{1, 2, \ldots, n\}$ and sends $A := g^a$ to Bob.
2. Bob samples $b \in \mathbb{R} \{1, 2, \ldots, n\}$ and sends $B := g^b$ to Alice.
3. Alice computes $k := B^a$ and Bob also computes $k := A^b$.

- Given $(g^a, g^b)$, for an eavesdropper, the distribution of the key $k = g^{ab}$ seems indistinguishable from the random element $g^r$.
- Alice and Bob can perform steps 1 and 2 simultaneously.
Any two-message key agreement protocol can be converted into a public-key encryption scheme

1. Gen(): Return a public key \( pk = A := g^a \) and a secret key \( sk = a \)

2. \( Enc_{pk}(m) \): Compute \( B := g^b \) and \( c := m \cdot A^b \). The ciphertext is \( (B, c) \)

3. \( Dec_{sk}(\tilde{B}, \tilde{c}) \): Compute \( \tilde{m} / (\tilde{B})^a \), where \( sk = a \).
Groups where DDH holds

1. The subgroup of $k$-th residues modulo a prime $p = k \cdot q + 1$, where $q$ is also a prime. When $k = 2$, it is quadratic residues modulo a safe prime

2. For a safe prime $p = 2 \cdot q + 1$, the quotient group $\mathbb{Z}_p^*/\{±1\}$

3. A prime-order elliptic curve over a prime field (with some additional technical restrictions)

4. A Jacobian of a hyper-elliptic curve over a prime field (with some additional technical restrictions)
Security Game for DDH.

1. The honest challenge samples a bit $u \in_R \{0, 1\}$
2. If $u = 0$, then it samples $(\alpha, \beta, \gamma)$ from the distribution $(g^a, g^b, g^{ab})$, where $a, b \in_R \{1, 2, \ldots, n\}$. If $u = 1$, then it samples $(\alpha, \beta, \gamma)$ from the distribution $(g^a, g^b, g^r)$, where $a, b, r \in_R \{1, 2, \ldots, n\}$
3. The honest challenge sends $(\alpha, \beta, \gamma)$ to the adversary
4. Adversary replies back with $\tilde{u} \in \{0, 1\}$ (its guess of the bit $u$)
5. The adversary wins the game if (and only if) $u = \tilde{u}$.
6. The DDH assumption states that any computationally efficient adversary only has a small (or, negligible) advantage in predicting the bit $u$
Suppose \((G, \times)\) be a group generated by \(g\), and discrete log is easy to compute. That is, given \(X := g^x\) as input, it is easy to compute \(x \in \{1, 2, \ldots, |X|\}\) (say, using an algorithm \(A\)).

Using such an algorithm, it is easy to construct a DDH adversary and break that assumption.

1. Our adversary receives \((\alpha, \beta, \gamma)\) from the honest challenger
2. Feeds \(\alpha\) as input to the algorithm \(A\) and recovers \(a\)
3. Compute \(\delta := \beta^a\)
4. If \(\gamma = \delta\), set \(\tilde{u} = 0\); otherwise, set \(\tilde{u} = 1\)

Food for thought: Compute the advantage of our adversary

The contrapositive of this statement is that if DDH is hard for a group, then DL is also hard for that group.
Suppose there is an algorithm that, given $X = g^x$ as input, can determine whether $x$ is even or not.

Note that when $\gamma = g^{ab}$, the exponent $ab$ is even with probability $3/4$.

However, when $\gamma = g^r$, the exponent $r$ is even with probability $1/2$.

So, using the algorithm mentioned above, we can construct an adversary who has a constant advantage in predicting $u$.

Food for thought: Construct this adversary and compute its distinguishing advantage.
The computational Diffie-Hellman assumption (CDH) states that given \((g^a, g^b)\), where \(a, b \in_R \{1, 2, \ldots, n\}\), it is computationally inefficient to compute \(g^{ab}\).

Note that if CDH is easy in a group, then there is an algorithm to compute \(g^{ab}\) from \((g^a, g^b)\). In this group, using this algorithm, an adversary can show that DDH is easy.

The contrapositive of this statement is that if DDH is hard for a group, then CDH is also hard for that group.