1. CDH and DDH

One of the most important goals in Cryptography is to identify the exact complexity assumptions used by cryptographic protocols.

CDH is where the Modern Cryptography originated.

CDH implies difficulty of computing discrete logarithms. The converse, however, is unknown for most practical groups.

CDH by itself is not sufficient to prove that the Diffie-Hellman protocol is useful for practical cryptographic purposes.

There is a theoretical way to use CDH alone: we can derive one unpredictable bit (known as a hard core bit) from g^{ab}. If, given g^a, g^b, the adversary could predict the hard core bit of g^{ab}, they could also compute g^{ab} in its entirety.

Later a much stronger assumption was introduced. The Decisional Diffie-Hellman (DDH) assumption:
given the values g^a and g^b, one can’t efficiently distinguish g^{ab} from a random group element.

We refer to groups in which the DDH assumption holds as DDH groups.

The need to rely on the DDH disqualifies many natural groups where the assumption does not hold. For example, any group whose order is divisible by small factors, such as the classic groups Z_p^*, where CDH is believed to hold (distinguishing can be based on Legendre symbol of g^{ab}).

Examples of groups in which DDH is believed to be intractable.

1. Let $p = 2q + 1$ where both p and q are prime, and let Q_q be the subgroup of quadratic residues in Z_p^*. It is a cyclic group of prime order. The family of groups is parameterized by p.

2. More generally, let $p = aq + 1$ where both p and q are prime and $q > p^{1/10}$. Let $Q_{p,q}$ be the subgroup of Z_p^* of order q. This family of groups is parameterized by both p and q.

3. Let $N = pq$ where p, q, $(p-1)/2$, $(q-1)/2$ are prime. Let T be the cyclic subgroup of Z_N^* of order ¼ $(p-1)(q-1)$. Although T does not have prime order, DDH is believed to be intractable. The group family is parameterized by N.

4. Let p be a prime and $E_{a,b}/F_p$ be an elliptic curve where $\mathrm{ord}(E_{a,b})$ is prime. The group is parameterized by (p, a, b).

We'll discuss certain results on the security of DDH later.
In the cryptographic literature, it is often recommended to work over subgroups of large prime order where no attacks are known on the DDH assumption.

2. DDH may not be sufficient

The DDH assumption, while apparently necessary, may be insufficient for guaranteeing the security of even the most basic DH transform applications.

Consider the ElGamal encryption scheme: given a public key $y = g^a$ (for secret a), a message $m \in G$ is encrypted by the pair (g^b, my^b) where the value b is chosen randomly anew for each encryption.

DDH guarantees the semantic security of the scheme (against chosen-plaintext attacks) provided that the plaintexts m are elements of the group G. However, if the message space is different, e.g., the set of strings of some length smaller than log(ord(G)), then the above encryption scheme becomes problematic.
We need to encode messages m as group elements in G, which is not straightforward when G is, e.g., a subgroup of prime order of Z_p^*. A naive (and common) approach of encoding m as an integer and performing the multiplication my^b modulo p makes the scheme insecure even if the group G satisfies DDH.

A good illustration of the potential weaknesses of this straightforward (or “textbook") application of ElGamal is presented in [Boneh, Joux, Nguyen, “Why Textbook ElGamal and RSA Encryption are Insecure”]. They show that if the space of plaintexts consists of random strings of length shorter than log(ord(G)) (e.g., when using public key encryption to encrypt symmetric keys), the above scheme turns out to be insecure even under a ciphertext-only attack and even if the group G is DDH.

Here is one of the [BJN] results:

Suppose the plaintext m is k bits long.
Let <p, g, y> be an ElGamal public key. When the order of g is at most $p/2^k$, it is possible to recover m from any ElGamal ciphertext of m in the time it takes to compute $2^{k/2+1}$ modular exponentiations. The attack succeeds with probability 18% (over the choice of m from $\{0, 1, \ldots, 2^k-1\}$), and requires k $2^{k/2}$ bits of memory. Can be parallelized.

(A meet-in-the-middle method based on the fact that a relatively small integer can often be expressed as a product of much smaller integers.)

A general and practical approach to addressing the above problems:
instead of using the DH value itself to “mask" m via multiplication, we hash the DH value g^{ab} to obtain a pseudorandom key K of suitable length, then use K to encrypt m under a particular encryption function (e.g., one-time pad).

The hash function here is used to extract the pseudorandomness present in the DH value. Universal Hash Functions, for instance, possess the required extraction properties. This is common to many applications of the DH transform including the Diffie-Hellman key-exchange protocol, where we derive agreed shared keys via hashing of the DH result.

3. And if we have to hash anyway?

Can we relax our requirements? Can we work over non-DDH groups? In particular, is doing hashed DH over Z_p^* secure?

The main result, informally:
For any cyclic group G, applying the hashed DH transform over G has the same security as applying the hashed DH transform directly over the maximal disjoint DDH subgroup of G.

So, we are only concerned with the existence of a sufficiently large DDH subgroup (no need to know its exact size or structural properties, nor how to construct it). In the Z_p^* case, it is enough to assume that DDH holds on large prime-order subgroups of Z_p^* and we can work directly over Z_p^*, where p is an unconstrained random prime.

Machinery:

t-DDH assumption (as a relaxation of DDH).
Informally, a group *G* satisfies the *t*-DDH assumption ($0 < t \leq \log(\mathrm{ord}(G))$) if given the pair ($g^a$, g^b), the value g^{ab} contains *t* bits of computational entropy.

Then the entropy-smoothing theorem gives us a way to efficiently transform (via universal hashing) DH values over groups in which the *t*-DDH assumption holds into shorter outputs that are computationally indistinguishable from the uniform distribution.

To be 2^{-k}-computationally close to uniform one can output up to (t - $2k$) pseudorandom bits (e.g., to produce 128-bit keys with a security parameter of $k = 80$, the group *G* should be 288-DDH).
We show that if *G* contains a DDH subgroup of order *m*, then *G* is log(*m*)-DDH.

Direct product characterization of the DDH assumption: we show that a group is DDH if and only if it is the direct product of (disjoint) prime power DDH subgroups.

In particular, this result plays a central role in our proof that the hashed DH transform over Z_p^* is secure as long as the DDH assumption holds in the subgroups of Z_p^* of large prime order.

Short-Exponent Diffie-Hellman. Can one use short exponents (e.g. as in [RFC2409]) and still preserve the security of the hashed DH transform? An obviously necessary requirement for the short exponent practice to be secure is the assumption that the discrete log problem is hard when exponents are restricted to a short length (say of *s* bits). This requirement (called the *s*-DLSE assumption) is, in fact, sufficient. More precisely, we can prove that if the *s*-DLSE assumption holds in a group *G*, then the hashed DH transform in *G* is as secure with full exponents as with *s*-bit exponents.

Immediate practical impact: the results justify certain practices in IKE and other commonly used protocols.

4. Discussing formalities

Probability ensembles $\{D_n\}$, with each distribution D_n is taken over a set $A_n \subset \{0, 1\}^{n'}$, where *n'* is polynomial in *n* (each ensemble has a fixed polynomial in *n* that determines the value *n'*).

Statistical and computational indistinguishability:

Let X_n, Y_n be two probability distributions over a support set A_n. We say that X_n and Y_n have statistical distance bounded by $d(n)$ if

$$\sum_{x \in An} |\Pr_{Xn}(x) - \Pr_{Yn}(x)| \leq d(n)$$

We say that the ensembles X_n and Y_n are statistically indistinguishable if for every polynomial $P(\cdot)$ and for all sufficiently large *n* we have that $d(n) \leq 1/ P(n)$.

We say that the probability ensembles X_n and Y_n are computationally indistinguishable (by non-uniform distinguishers) if no polynomial size circuit (family) can distinguish between samples drawn according to X_n or according to Y_n.

[GKR]'s choice: asymptotic model, non-uniform distinguishers, "security of individual groups".

We consider infinite families of cyclic groups $\{(G_n, g_n, m_n)\}_n$, where $\log(m_n)$ is bounded by a polynomial in n.

CDH problem: given a pair (g_n^a, g_n^b), compute g_n^{ab}. If this problem is intractable over a given group family, we say the CDH holds over the family.

DDH is a much stronger, but also more useful, assumption. Consider the family of sets G_n x G_n x G_n, and the following two probability ensembles:

$R_n = \{(g_n^a, g_n^b, g_n^c)$ for $a, b, c \in_R [0, m_n)\}$

and

$DH_n = \{(g_n^a, g_n^b, g_n^{ab})$ for $a, b \in_R [0, m_n)\}$

We say that DDH holds over a group family if the ensembles R_n and DH_n are computationally indistinguishable (with respect to non-uniform distinguishers).

Why non-uniform setting? We want to allow for “auxiliary input" for each group G_n in the family G being given to a distinguisher. [GKR]: “In formal terms one may assume that we have a single group G_n for each value of the security parameter n. This approach allows us to keep the simplicity of arguments in the asymptotic polynomial-time model while capturing the fact that we are interested in the security of individual groups for which the attacker may have some side information.”

Problems with “a single group G_n for each value of the security parameter n”.

The very first theorem – “forget asymptotic”, needs to be interpreted correctly.

Possible way to address the problems: talk of “advantages”, concrete security. We would then define (t, ε)-DDH and talk about individual (or finite collections of) groups and their subgroups.

Alternative definition – Instance Generator:

A group family G is a set of finite cyclic groups $\{G_\alpha\}$, where α ranges over an infinite index set. We assume there is a polynomial time (in $\log(\alpha)$) algorithm that given α and two elements in G_α outputs their sum.

An Instance Generator for G is a randomized algorithm that given an integer n (in unary), runs in polynomial time in n and outputs some random index α and a generator g of G_α. Note that for each n, the Instance Generator induces a distribution on the set of indices α.

The index α encodes the group parameters (e.g., (p, a, b) for $E_{a,b}/F_p$). The instance generator is used to select a random member of G of the appropriate size. For instance, when G is the family of prime order subgroups of Z_p^*, the instance generator, on input n, may generate a random n-bit prime p such that $(p-1)/2$ is also prime. In some cases it may make sense to generate distributions other than uniform. For instance, one may wish to avoid primes of the form $2^k + 1$.

In defining advantage of a distinguisher, we take probability over the random choice of (α, g) according to the distribution induced by the Instance Generator, the random choice of $a, b, c \in_R [0, \mathrm{ord}(G_\alpha))$, and random bits used by the distinguisher.

Which approach is a better model of real applications? Comes down to when (G, g) are chosen.

5. Some evidence about DDH security and an interesting DDH application

(a) Random self-reducibility (for groups of prime order): a distinguisher with a non-negligible advantage can be converted into an almost perfect one, that is, for a negligible function $\varepsilon(n)$ for large enough n, we have

$$\Pr [A(\alpha, g, g^a, g^b, g^c) = \text{“true”} \mid c = ab] > 1 - \varepsilon$$

$$\Pr [A(\alpha, g, g^a, g^b, g^c) = \text{“true”} \mid c \neq ab] < \varepsilon$$

Given a triple (x, y, z), we pick random integers u, v, w, and construct the triple $(x', y', z') = (x^w g^u, yg^v, z^w y^u x^{vw} g^{uv})$. If (x, y, z) is a valid DH triple, then (x', y', z') is a random DH triple, and if (x, y, z) is not a valid DH triple, then (x', y', z') is a random triple.
Then we generate k independent triples (x', y', z') as above and feed them to our DDH oracle, and then we generate k random triples and also feed them to the DDH oracle, and we count the number of “true” answers in the first and second experiments. If the difference is greater than $\delta k/2$ (where δ is our DDH oracle advantage), we output “true”, otherwise we output “false”. Knowledge of δ issue: either existential meaning or non-uniformity.

(b) Generic algorithms must make oracle queries to compute in given groups, they have to deal with random encodings (can't take advantage of a particular group encoding). Generic algorithms can't break DDH faster than breaking DL, and the complexity of the latter is provably lower bounded by $O(p^{1/2})$.

(c) Ability to compute $O(\log(p)^{1/2})$ most significant bits of g^{ab} implies the ability to compute the entire g^{ab}. (Lattice basis reduction and LLL.)

(d) Statistical distribution of DH triples in Z_p^*. Weyl discrepancy is small:

if B is a box in Z_p^3, then

$\sup_B | N(B) – \mathrm{vol}(B) (p\text{-}1)^2 / (p\text{-}1)^3 | = o(p^2)$

Proving a link between DDH and a known hard problem is a crucial open problem.
Another open problem: in a prime order subgroup of Z_p^*, is there an algorithm for DDH better than the fastest DL algorithm in that subgroup?

An interesting application of DDH

Naor and Reingold ("Number theoretic constructions of efficient pseudo random functions") describe a beautiful application of DDH. Prior to their results, existing constructions based on number theoretic primitives were by far less efficient.

At a high level, a set F_n of functions $A_n \to B_n$ (finite domains) is called a pseudo random function ensemble if no efficient statistical test can distinguish between a random function chosen in the set and a truly random function, i.e. a function chosen at random from the set of all functions $A_n \to B_n$. The statistical test is only given "black-box" access to the function. That is, it can ask an oracle to evaluate the given function at a point of its choice, but cannot view the internal implementation.

Let G be a group family. In the NR construction, F_n is a set of functions from $\{0, 1\}^n$ to G_α (index α may be different for different functions in F_n). A function from F_n is parameterized by a triple (α, g, a), where the first two values come from the Instance Generator, and the third one is a vector of (n+1) random integers in the range [1, ord(G_α)). Given $x = x_1 \ldots x_n$, we compute

$$f_{(\alpha, g, a)}(x) = g^{a_0 a_1^{x1} \ldots a_n^{xn}}$$

Such functions can be evaluated very efficiently.

The main result is:

Let A^f be the algorithm A with access to an oracle for evaluating f. Let G be a group family and suppose DDH holds for G. Let $\{F_n\}$ be the NR pseudo-random function ensemble. Then for every probabilistic polynomial-time algorithm A and sufficiently large n, we have

$$| \Pr(A^f(\alpha, g) = \text{“true”}) - \Pr(A^R(\alpha, g) = \text{“true”}) | < \varepsilon$$

for a negligible function $\varepsilon(n)$. The first probability is over the choice of f from F_n. The second probability is over the random distribution induced by the Instance Generator and the random choice of the function R among the set of all $\{0, 1\}^n \rightarrow G_\alpha$ functions.

6. Direct Product DDH Characterization

Lemma 1 If DDH holds in a cyclic group G, then it holds in all the subgroups of G.

Lemma 2 Let G be a cyclic group of order $m = m_1m_2$, where $(m_1, m_2) = 1$, and let G_1 and G_2 be the subgroups of orders m_1, m_2 respectively. If DDH holds in G_1 and G_2, then it holds in G.
(Lift pairs of triples in G_1 and G_2 into G and consider hybrid distributions.)

How can this be interpreted asymptotically? Consider a family $\{G_n\}$ and the family of all prime power order subgroups of groups in G (indexed in a way consistent with indexing in G). Then DDH holds in one family iff it holds in the other one.

Why is this called "Direct Product Characterization"?

Enjoying freedom in choosing consistent generators in G and its subgroups? Non-uniform model helps: if DDH holds for one generator of a group, then it holds for all generators. This is because if for some generator g DDH is easy, then for any other generator h, we simply provide the algorithm with $\log_g(h)$ as an auxiliary input.

Do we really need $(m_1, m_2) = 1$ condition in Lemma 2?

There is at least an example by Don Coppersmith of a cyclic group G of order q^2 which contains a subgroup H of order q, such that H is believed to be DDH while G is trivially not DDH (can be constructed for any given DDH group of order q).

Coppersmith's example:

Given a cyclic DDH group H of order q with generator g, we take $G = \{(h, a) \colon h \in H, a \in Z_q\}$, and the group operation * is defined as $(h_1, a_1) * (h_2, a_2) = (h, a)$, where
(i) if $a_1 + a_2 < q$ then $h = h_1 h_2$, $a = a_1 + a_2$; and
(ii) $h = g h_1 h_2$, $a = a_1 + a_2 - q$ otherwise.

G is a cyclic group with generator $(e, 1)$.

Clearly, $(e, 1)^q = (g, 0)$ generates the DDH subgroup H x $\{0\}$. However, G is not even CDH.

We may have a better luck in some specific families of groups. An important open question: how plausible is DDH in prime-power order subgroups of Z_p^*?

7. The *t*-DDH assumption. Large DDH subgroups and *t*-DDH in ambient groups.

Min-entropy: $\mathrm{H}_{\min}(X) = \min_{x:\Pr(x)\neq 0}(-\log(\Pr(x)))$

Universal Hash Functions:

Definition Let H_n be a family of functions, where each $h \in H_n$ is defined as $h: A_n \to \{0, 1\}^{m(n)}$. We say that H_n is a family of (pairwise-independent) universal hash functions if, for all $x, x' \in A_n$, $x \neq x'$, and for all $a, a' \in \{0, 1\}^{m(n)}$ we have
$\Pr_{h\in Hn} [h(x) = a \text{ and } h(x') = a'] = 2^{-2m(n)}$.
That is, a randomly chosen h will map any pair of distinct elements independently and uniformly.

Entropy Smoothing Theorem

Theorem Let t be a positive integer and let X be a random variable defined on $\{0, 1\}^n$ such that $\mathrm{H}_{\min}(X) > t$. Let $k > 0$ be an integer parameter. Let H be a family of universal hash functions such that $h \in H$, $h: \{0, 1\}^n \to \{0, 1\}^{t-2k}$. Let U be the uniform distribution over $\{0, 1\}^{t-2k}$. Then, the distributions $<h(X), h>$ and $<U, h>$, where $h \in_{\mathrm{R}} H$, have statistical distance at most $2^{-(k+1)}$.

Definition A probability ensemble Y_n has computational entropy $t(n)$ if there exists a probability ensemble X_n such that $\mathrm{H}_{\min}(X_n) \geq t(n)$ and X_n and Y_n are computationally indistinguishable.

Using a standard hybrid argument it is easy to show that the Entropy Smoothing Theorem, as discussed above, can be generalized to probability ensembles X_n that have computational entropy $t(n)$. In this case, applying a (randomly chosen) universal hash function X_n results in a pseudorandom ensemble, namely, an ensemble which is computationally indistinguishable from the uniform distribution.

Definition We say that the $t(n)$-DDH Assumption holds over a group family $G = \{G_n\}$ if for all n there exists a family of probability distributions $X_n(g_n^a, g_n^b)$ over G_n (one distribution for each pair g_n^a, g_n^b) such that:

$\mathrm{H}_{\min}(X_n(g_n^a, g_n^b)) \geq t(n)$ and

the probability ensemble DH_n is computationally indistinguishable from the ensemble $R_n^* = \{(g_n^a, g_n^b, C)$ for $a, b \in_{\mathrm{R}} [0, \mathrm{ord}(G_n))$ and $C \in X_n(g_n^a, g_n^b)\ G_n\}$.

It is important to note that the distributions $X_n(g_n^a, g_n^b)$ in the above definition may be different for each pair of values (g_n^a, g_n^b). Requiring instead a single distribution X for all pairs (as may seem more natural at the first glance) results in a significantly stronger, and consequently less useful, assumption.

Reconsider the example with breaking DDH in Z_p^* by inspecting Legendre symbols.

DDH assumption over a group G is equivalent to the t-DDH assumption over G for $t = \log(\mathrm{ord}(G))$.

Samplable (semi-samplable) t-DDH assumption: we require $X_n(g_n^a, g_n^b)$ to be polynomial-time samplable (resp., polynomial-time samplable when either exponent a or b is known). Will need that only when studying DDH with short exponent.

If the t-DDH assumption holds, we can apply universal hashing to DH values g^{ab} and obtain distribution, which is computationally indistinguishable from the uniform distribution even when the hash function and both g^a and g^b are given to the distinguisher.

For groups of prime order, the t-DDH Assumption is equivalent to the full DDH assumption:

Lemma Let G be a group of prime order q. If the t-DDH assumption holds in G for $t \geq 1$, then the DDH assumption holds in G as well.
(Upper bound $\Pr(D(x) = 1)$ for random x, lower bound $\Pr(D(x) = 1)$ for x chosen in accordance with R^*.)

So we have an interesting all-or-nothing law for prime order groups. Can we show that CDH is equivalent to DDH for such groups?

Theorem Let G be a cyclic group of order $m = m_1 m_2$ where $(m_1, m_2) = 1$, and G_1 be a subgroup of order m_1 in G. If the DDH assumption holds over G_1 then the $\log(m_1)$-DDH assumption holds in G.
(Choose i, j at random in Z_m, construct $A = A_1 g^i$, $B = B_1 g^j$, …)

8. Working over Z_p^*.

If our application requires a pseudorandom output of l bits and our security parameter is k, the above results show that the hashed DH is secure over Z_p^* provided that (p-1) has enough prime divisors (with multiplicity 1) whose product is larger than the entropy bound 2^{l+2k}, and for which the subgroups of corresponding prime order are DDH. In particular, the fact that (p-1) has additional smaller prime factors does not invalidate the security of the hashed DDH in Z_p^*.

A particularly interesting group is Z_p^* for $p = 2q + 1$, q prime, because it is ($\log p - 1$)-DDH under the standard assumption. Note that these groups are free from the potentially serious attacks described in [Lim, Lee, "A Key Recovery Attack on DL-Based Schemes Using a Prime Order Subgroup"] that affect subgroups of prime order q where (p-1)/q has a relatively large smooth factor.

Usually, one has to know the full or partial factorization of $(p - 1)$, which is essential for selecting a generator of the group. It is a theoretically and practically important to establish whether the knowledge of the factorization of $(p - 1)$ is essential for working securely over Z_p^* or over one of its subgroups. We show that the answer is negative. Specifically, it follows from our results that if one chooses a random prime p (of an appropriate size) and a random element g in Z_p^*, then performing the hashed DH transform over the group generated by g is secure (based on heuristic facts about large numbers factorization, believed by most computational number theorists).

Let p be a random prime such that $p - 1 = p_1 p_2 \cdots p_n$ and $p_1 \leq p_2 \leq \ldots \leq p_n$ are all (not necessarily different and possibly unknown) primes. Let h be an element randomly chosen from Z_p^*, and let G_h denote the subgroup generated by h. We first claim that with overwhelming probability the large prime factors of $(p - 1)$ divide the order of G_h. Namely:

Lemma For all $1 \leq i \leq$ n: $\Pr(p_i$ doesn't divide $\mathrm{ord}(h)) \leq 1/ p_i$.

Corollary If $p_j, \ldots, p_n > B$, for a given bound B, then
$\Pr(p_j \cdots p_n \mid \mathrm{ord}(h)) \geq 1 - (1/ p_j + \ldots + 1/ p_n) \underset{\cdot}{\geq} 1 - (n - j + 1)/B \geq 1 - \log(p)/B$

Thus, for large values of B, with overwhelming probability, G_h has as subgroups all the prime-order subgroups of Z_p^*whose order is larger than B. If we denote by P the product of all prime factors of $(p - 1)$ larger than B, and assume that the DDH holds in subgroups of prime order larger than B, we get that G_h is $\log(P)$-DDH.

All that is left to argue is that log(P) is large enough. For this we use the following lemma providing an upper bound on the expected size of the product of all prime divisors of (p – 1) that are smaller than B (which gives a lower bound on the expected size of log(P)).

Lemma For a random prime p (as above) and a fixed bound B, the expected length of $\prod p_i$, where $p_i < B$, is log(B) + 1.

In other words, the lemma states that the expected size of log(P) is log(p) – log(B).

To illustrate, if we set log(p) = 1024 and log(B) = 160, then we expect G_h to be 864-DDH. However, it's just an expected value, and if all that we have is Markov inequality, our estimates will not be very impressive. If p happens to have a B-smooth part that is 4 times larger than expected, we are still left with a 384-DDH subgroup G_h with enough computational entropy for most DH applications (such as deriving a 128-bit pseudorandom key), which sounds good. But the probability of the B-smooth part being 6 times larger than expected is not that small. Upper bounding the variance?