Negotiated Discrete Log DHE Groups

Status Quo

- **Client**: I want to use discrete-log-based DHE
- **Server**: OK, here is a group: g, p, and here is my share b

Problems

- Client cannot cheaply evaluate the quality of the defined group:
  - is *p* actually prime?
  - does *g* avoid generating a small subgroup?
- Client cannot indicate to server the preferred strength
- Client's only option when shown a bad group is to abort
- Server does not know what groups client can handle
  - Java before Java 8 can't do > 1024-bit DL DHE :(

Proposal `draft-ietf-tls-negotiated-dl-dhe`

- Establish a registry of named DL DHE groups and a way to negotiate them

Connection now looks like:

- **Client**: I want to use discrete-log-based DHE, my preferred groups are `dldhe3072`, `dldhe4096`, `dldhe2432`
- **Server**: OK, we'll use `dldhe3072`, and here is my share *b*

Similar to named curves for EC DHE groups.

Advantages

- Collective vetting of the named DL DHE groups
- Clear capability indication
- Shorter handshakes
- Precomputation for implementations
- Given known groups, short exponents possible (e.g. §6.2 of `RFC 4419` (ssh))

Proposed registry

The proposed registry uses the same form for rigidity as `RFC 3526` , but using `e` as the fundamental constant instead of `π`

For a given bitlength `b`, find the lowest positive integer `X` that creates a safe prime `p` where:

```
p = 2^b - 2^{b-64} + {[2^{b-130} e] + X } * 2^64 - 1
```

In practice: 64 bits of 0xFF || `e` + `X`, || 64 bits of 0xFF

```
p = FFFFFFFF FFFFFFFF ADF85458 A2BB4A9A AFDC5620 ...
    ...      FFFFFFFF FFFFFFFF
```

Currently: `dldhe2432`, `dldhe3072`, `dldhe4096`, `dldhe6144`, `dldhe8192`

Open Issues

- Should we share known groups with other systems, like IKE (e.g. `RFC 3526`) or SRP?
- Should handshake record type be `ServerDHParams` or some new handshake type?