Diffie–Hellman reductions

Lorenz Panny

Technische Universiteit Eindhoven

COSIC seminar, Leuven, 2 August 2019

Hardness reductions in cryptography

Recall RSA encryption (simplified special case):

- Private key: two big prime numbers p, q.
- Public key: their product $n = pq$.
- Encrypt: compute $c = m^{65537} \bmod n$.
- Decrypt: compute $m = c^{65537^{-1} \bmod \operatorname{lcm}(p-1, q-1)} \bmod pq$.

Hardness reductions in cryptography

Recall RSA encryption (simplified special case):

- Private key: two big prime numbers p, q.
- Public key: their product $n = pq$.
- Encrypt: compute $c = m^{65537} \bmod n$.
- Decrypt: compute $m = c^{65537^{-1} \bmod \operatorname{lcm}(p-1, q-1)} \bmod pq$.

Clearly, anyone who can factor n can decrypt.
Q: Can everyone capable of decrypting also factor n?

Hardness reductions in cryptography

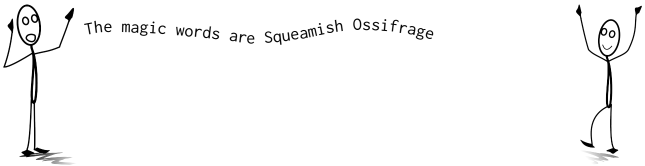
Recall RSA encryption (simplified special case):

- Private key: two big prime numbers p, q.
- Public key: their product $n = pq$.
- Encrypt: compute $c = m^{65537} \bmod n$.
- Decrypt: compute $m = c^{65537^{-1} \bmod \operatorname{lcm}(p-1, q-1)} \bmod pq$.

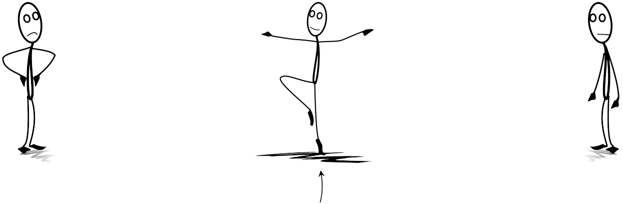
Clearly, anyone who can factor n can decrypt.
Q: Can everyone capable of decrypting also factor n?

If yes:
No point attacking RSA specifically; just focus on factoring.

Diffie–Hellman key exchange (very big picture)



Diffie–Hellman key exchange (very big picture)

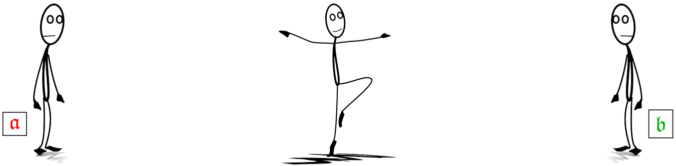


evil eavesdropper Eve!

Diffie–Hellman key exchange (very big picture)

Diffie–Hellman key exchange (very big picture)

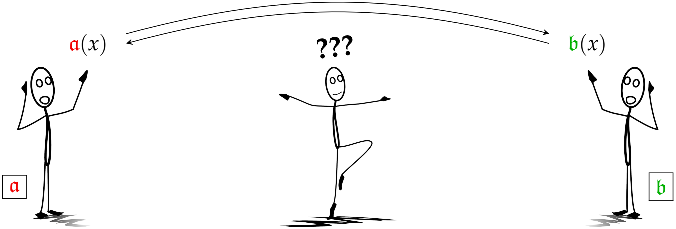
Parameters: a finite set X, a fixed element $x \in X$.



- Private keys:
  efficient functions $\mathfrak{a}, \mathfrak{b} \colon X \to X$.

Diffie–Hellman key exchange (very big picture)

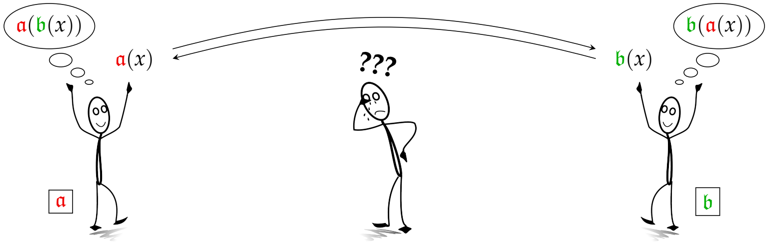
Parameters: a finite set X, a fixed element $x \in X$.



- Private keys:
  efficient functions $\mathfrak{a}, \mathfrak{b} \colon X \to X$.
- Public keys: the elements $\mathfrak{a}(x), \mathfrak{b}(x) \in X$.

Diffie–Hellman key exchange (very big picture)

Parameters: a finite set X, a fixed element $x \in X$.



- <u>Private</u> keys:
  efficient functions $\mathfrak{a}, \mathfrak{b} \colon X \to X$ such that $\mathfrak{a} \circ \mathfrak{b} = \mathfrak{b} \circ \mathfrak{a}$.
- <u>Public</u> keys: the elements $\mathfrak{a}(x), \mathfrak{b}(x) \in X$.
- <u>Shared</u> secret: the element $\mathfrak{a}(\mathfrak{b}(x)) = \mathfrak{b}(\mathfrak{a}(x))$.

This talk

Is computing $\mathfrak{a}(\mathfrak{b}(x))$ as hard as recovering $\mathfrak{a}$ or $\mathfrak{b}$?

This talk

Is computing $\mathfrak{a}(\mathfrak{b}(x))$ as hard as recovering $\mathfrak{a}$ or $\mathfrak{b}$?

- Standard proof technique:
  Use a black-box 'oracle'

$$\mathcal{O}: \ (x, \mathfrak{a}(x), \mathfrak{b}(x)) \longmapsto \mathfrak{a}(\mathfrak{b}(x))$$

This talk

Is computing $\mathfrak{a}(\mathfrak{b}(x))$ as hard as recovering $\mathfrak{a}$ or $\mathfrak{b}$?

- Standard proof technique:
  Use a black-box 'oracle'

$$\mathcal{O}\colon\ (x, \mathfrak{a}(x), \mathfrak{b}(x)) \longmapsto \mathfrak{a}(\mathfrak{b}(x))$$

  to construct an efficient algorithm

$$\mathcal{A}(\mathcal{O})\colon\ \mathfrak{a}(x) \longmapsto \mathfrak{a}\,.$$

This talk

Is computing $\mathfrak{a}(\mathfrak{b}(x))$ as hard as recovering $\mathfrak{a}$ or $\mathfrak{b}$?

- Standard proof technique:
  Use a black-box 'oracle'
  $$\mathcal{O}\colon\ (x, \mathfrak{a}(x), \mathfrak{b}(x)) \longmapsto \mathfrak{a}(\mathfrak{b}(x))$$
  to construct an efficient algorithm
  $$\mathcal{A}(\mathcal{O})\colon\ \mathfrak{a}(x) \longmapsto \mathfrak{a}\,.$$

Intuition: *Every* efficient $\mathcal{O}$ leads to an efficient $\mathcal{A}(\mathcal{O})$.

Group-based Diffie–Hellman

The only reasonable Diffie–Hellman instantiations 1976–2017:
$(G, \cdot)$ a finite group; $\mathfrak{a}, \mathfrak{b}$ exponentiations.

Group-based Diffie–Hellman

The only reasonable Diffie–Hellman instantiations 1976–2017:
$(G, \cdot)$ a finite group; $\mathfrak{a}, \mathfrak{b}$ exponentiations.

- Private keys: $\mathfrak{a}, \mathfrak{b} \in \mathbb{Z}/\operatorname{ord} g$.
- Public keys: $g^{\mathfrak{a}}, g^{\mathfrak{b}}$.
- Shared secret: $(g^{\mathfrak{a}})^{\mathfrak{b}} = (g^{\mathfrak{b}})^{\mathfrak{a}} = g^{\mathfrak{a}\mathfrak{b}}$.

Group-based Diffie–Hellman

The only reasonable Diffie–Hellman instantiations 1976–2017:
$(G, \cdot)$ a finite group; $\mathfrak{a}, \mathfrak{b}$ exponentiations.

- Private keys: $\mathfrak{a}, \mathfrak{b} \in \mathbb{Z}/\operatorname{ord} g$.
- Public keys: $g^{\mathfrak{a}}, g^{\mathfrak{b}}$.
- Shared secret: $(g^{\mathfrak{a}})^{\mathfrak{b}} = (g^{\mathfrak{b}})^{\mathfrak{a}} = g^{\mathfrak{a}\mathfrak{b}}$.

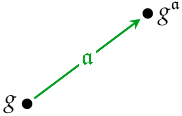
Examples:

- Multiplicative groups of finite fields $(\mathbb{F}_q^*, \cdot)$.
- Elliptic curves $E\colon y^2 = x^3 + Ax^2 + x$ with ‘weird’ addition.

Problems from Diffie–Hellman

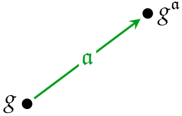
Problems from Diffie–Hellman

- <u>Discrete-logarithm problem (**DLP**)</u>
  Compute $\mathfrak{a}$ from $g, g^{\mathfrak{a}}$.

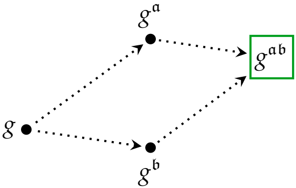


Problems from Diffie–Hellman

- <u>Discrete-logarithm problem (**DLP**)</u>
  Compute $\mathfrak{a}$ from $g, g^{\mathfrak{a}}$.



- <u>Computational Diffie–Hellman problem (**CDH**)</u>
  Compute $g^{\mathfrak{a}\mathfrak{b}}$ from $g, g^{\mathfrak{a}}, g^{\mathfrak{b}}$.



Generic complexity of DLP (Pohlig–Hellman 1978)

- Upshot: If the factorization of $|G|$ is $p_1^{e_1} \cdots p_r^{e_r}$, then one can solve DLP in $O\big(\sum_{i=1}^{r} e_i \cdot (\log|G| + \sqrt{p_i})\big)$ group operations.

  $\Longrightarrow$ Cost dominated by the biggest prime factor of $|G|$.

  $\Longrightarrow$ DLP is easy if $|G|$ is smooth (i.e., no big prime factors).

Generic complexity of DLP (Pohlig–Hellman 1978)

- Upshot: If the factorization of $|G|$ is $p_1^{e_1} \cdots p_r^{e_r}$, then one can solve DLP in $O\big(\sum_{i=1}^{r} e_i \cdot (\log|G| + \sqrt{p_i})\big)$ group operations.

  $\Longrightarrow$ Cost dominated by the biggest prime factor of $|G|$.

  $\Longrightarrow$ DLP is easy if $|G|$ is smooth (i.e., no big prime factors).

- This is a generic-group algorithm.
  It is irrelevant how G is represented or what $\cdot$ does.
  The algorithm only requires subroutines that compute $\cdot$ and $^{-1}$.

Generic complexity of DLP (Pohlig–Hellman 1978)

- Upshot: If the factorization of $|G|$ is $p_1^{e_1} \cdots p_r^{e_r}$, then one can solve DLP in $O\big(\sum_{i=1}^{r} e_i \cdot (\log |G| + \sqrt{p_i})\big)$ group operations.

  $\Longrightarrow$ Cost dominated by the biggest prime factor of $|G|$.

  $\Longrightarrow$ DLP is easy if $|G|$ is smooth (i.e., no big prime factors).

- This is a generic-group algorithm.
  It is irrelevant how G is represented or what $\cdot$ does.
  The algorithm only requires subroutines that compute $\cdot$ and ${}^{-1}$.

! Shoup 1997: This is essentially optimal for *generic* groups.

Generic complexity of DLP (Pohlig–Hellman 1978)

- Upshot: If the factorization of $|G|$ is $p_1^{e_1} \cdots p_r^{e_r}$, then one can solve DLP in $O\big(\sum_{i=1}^{r} e_i \cdot (\log|G| + \sqrt{p_i})\big)$ group operations.

  $\Longrightarrow$ Cost dominated by the biggest prime factor of $|G|$.

  $\Longrightarrow$ DLP is easy if $|G|$ is smooth (i.e., no big prime factors).

- This is a generic-group algorithm.
  It is irrelevant how G is represented or what $\cdot$ does.
  The algorithm only requires subroutines that compute $\cdot$ and $^{-1}$.

! Shoup 1997: This is essentially optimal for *generic* groups.

!! There are many groups where one can solve DLP faster.

Diffie–Hellman's algebraic properties

Anyone can...

- encode numbers x in the exponents: compute g^x.

Diffie–Hellman's algebraic properties

Anyone can...

- encode numbers x in the exponents: compute g^x.
- add in the exponents: $g^{\mathfrak{a}+\mathfrak{b}} = g^{\mathfrak{a}} \cdot g^{\mathfrak{b}}$.
- negate exponents: $g^{-\mathfrak{a}} = (g^{\mathfrak{a}})^{-1}$.

Diffie–Hellman's algebraic properties

Anyone can...

- encode numbers x in the exponents: compute g^x.
- add in the exponents: $g^{\mathfrak{a}+\mathfrak{b}} = g^{\mathfrak{a}} \cdot g^{\mathfrak{b}}$.
- negate exponents: $g^{-\mathfrak{a}} = (g^{\mathfrak{a}})^{-1}$.

Anyone who can solve CDH can...

- multiply exponents: $g^{\mathfrak{a}\cdot\mathfrak{b}} = \mathit{shared\_secret}(g, g^{\mathfrak{a}}, g^{\mathfrak{b}})$.

Diffie–Hellman's algebraic properties

Anyone can...

- encode numbers x in the exponents: compute g^x.
- add in the exponents: $g^{\mathfrak{a}+\mathfrak{b}} = g^{\mathfrak{a}} \cdot g^{\mathfrak{b}}$.
- negate exponents: $g^{-\mathfrak{a}} = (g^{\mathfrak{a}})^{-1}$.

Anyone who can solve CDH can...

- multiply exponents: $g^{\mathfrak{a}\cdot\mathfrak{b}} = \mathit{shared\_secret}(g, g^{\mathfrak{a}}, g^{\mathfrak{b}})$.
- exponentiate exponents: square-and-multiply using $\circlearrowleft$.

Diffie–Hellman's algebraic properties

Anyone can...

- encode numbers x in the exponents: compute g^x.
- add in the exponents: $g^{\mathfrak{a}+\mathfrak{b}} = g^{\mathfrak{a}} \cdot g^{\mathfrak{b}}$.
- negate exponents: $g^{-\mathfrak{a}} = (g^{\mathfrak{a}})^{-1}$.

Anyone who can solve CDH can...

- multiply exponents: $g^{\mathfrak{a}\cdot\mathfrak{b}} = \mathit{shared\_secret}(g, g^{\mathfrak{a}}, g^{\mathfrak{b}})$.
- exponentiate exponents: square-and-multiply using $\circlearrowleft$.
- invert exponents: $g^{1/\mathfrak{a}} = g^{\mathfrak{a}^{\varphi(|G|)-1}}$ if $\gcd(\mathfrak{a}, |G|) = 1$ using $\circlearrowleft$.

Black-box rings

- We interpret $g^{\mathfrak{a}}$ as labels for the hidden elements $\mathfrak{a}$.
- With a CDH oracle we can perform arbitrary ring operations $(+,-,\cdot,/)$ on these hidden representations.
- **Notation**: Write $\lceil \mathfrak{a} \rfloor$ for the hidden element $g^{\mathfrak{a}}$.

Black-box rings

- We interpret $g^{\mathfrak{a}}$ as labels for the hidden elements $\mathfrak{a}$.
- With a CDH oracle we can perform arbitrary ring operations $(+, -, \cdot, /)$ on these hidden representations.
- **Notation**: Write $\lceil \mathfrak{a} \rfloor$ for the hidden element $g^{\mathfrak{a}}$.

The elements $g^{\mathfrak{a}}$ form a black-box ring isomorphic to $\mathbb{Z}/\operatorname{ord} g$.

Black-box rings

- We interpret $g^{\mathfrak{a}}$ as labels for the hidden elements $\mathfrak{a}$.
- With a CDH oracle we can perform arbitrary ring operations $(+, -, \cdot, /)$ on these hidden representations.
- **Notation**: Write $\lceil \mathfrak{a} \rfloor$ for the hidden element $g^{\mathfrak{a}}$.

The elements $g^{\mathfrak{a}}$ form a black-box ring isomorphic to $\mathbb{Z}/\operatorname{ord} g$.

We mostly care about black-box fields:
For discrete logarithms, it's sufficient to consider prime-order g.

First result: den Boer (1988)

Let $G = \mathbb{F}_p^*$, write $R = \mathbb{Z}/|G| = \mathbb{Z}/(p-1)$,
and suppose $|R^*| = \varphi(p-1)$ is smooth.
Then CDH is polynomial-time equivalent to DLP in $\mathbb{F}_p^*$.

First result: den Boer (1988)

Let $G = \mathbb{F}_p^*$, write $R = \mathbb{Z}/|G| = \mathbb{Z}/(p-1)$,
and suppose $|R^*| = \varphi(p-1)$ is smooth.
Then CDH is polynomial-time equivalent to DLP in $\mathbb{F}_p^*$.

Proof idea:
Solve a DLP in the exponents R^* to find a representation of $\lceil \mathfrak{a} \rfloor$ as a power of some known $\lceil \mathfrak{h} \rfloor$, then recompute $\mathfrak{a}$ in the clear.

First result: den Boer (1988)

Let $G = \mathbb{F}_p^*$, write $R = \mathbb{Z}/|G| = \mathbb{Z}/(p-1)$,
and suppose $|R^*| = \varphi(p-1)$ is smooth.
Then CDH is polynomial-time equivalent to DLP in $\mathbb{F}_p^*$.

Proof idea:
Solve a DLP in the exponents R^* to find a representation of $\lceil \mathfrak{a} \rfloor$ as a power of some known $\lceil \mathfrak{h} \rfloor$, then recompute $\mathfrak{a}$ in the clear.

Proof:

- Suppose (for simplicity) that R^* is cyclic with a generator $\mathfrak{h}$.

First result: den Boer (1988)

Let $G = \mathbb{F}_p^*$, write $R = \mathbb{Z}/|G| = \mathbb{Z}/(p-1)$,
and suppose $|R^*| = \varphi(p-1)$ is smooth.
Then CDH is polynomial-time equivalent to DLP in $\mathbb{F}_p^*$.

Proof idea:
Solve a DLP in the exponents R^* to find a representation of $\lceil\mathfrak{a}\rfloor$ as a power of some known $\lceil\mathfrak{h}\rfloor$, then recompute $\mathfrak{a}$ in the clear.

Proof:

- Suppose (for simplicity) that R^* is cyclic with a generator $\mathfrak{h}$.
- Encode $\mathfrak{h}$ to a black-box element $\lceil\mathfrak{h}\rfloor$ of R.

First result: den Boer (1988)

Let $G = \mathbb{F}_p^*$, write $R = \mathbb{Z}/|G| = \mathbb{Z}/(p-1)$,
and suppose $|R^*| = \varphi(p-1)$ is smooth.
Then CDH is polynomial-time equivalent to DLP in $\mathbb{F}_p^*$.

Proof idea:
Solve a DLP in the exponents R^* to find a representation of $\lceil \mathfrak{a} \rfloor$ as a power of some known $\lceil \mathfrak{h} \rfloor$, then recompute $\mathfrak{a}$ in the clear.

Proof:

- Suppose (for simplicity) that R^* is cyclic with a generator $\mathfrak{h}$.
- Encode $\mathfrak{h}$ to a black-box element $\lceil \mathfrak{h} \rfloor$ of R.
- Solve the DLP $(\lceil \mathfrak{h} \rfloor, \lceil \mathfrak{a} \rfloor)$ in the hidden version of R^* using Pohlig–Hellman. We get $k \in \mathbb{Z}$ such that $g^{\mathfrak{a}} = g^{\mathfrak{h}^k}$.

First result: den Boer (1988)

Let $G = \mathbb{F}_p^*$, write $R = \mathbb{Z}/|G| = \mathbb{Z}/(p-1)$,
and suppose $|R^*| = \varphi(p-1)$ is smooth.
Then CDH is polynomial-time equivalent to DLP in $\mathbb{F}_p^*$.

Proof idea:
Solve a DLP in the exponents R^* to find a representation of $\lceil \mathfrak{a} \rfloor$ as a power of some known $\lceil \mathfrak{h} \rfloor$, then recompute $\mathfrak{a}$ in the clear.

Proof:

- Suppose (for simplicity) that R^* is cyclic with a generator $\mathfrak{h}$.
- Encode $\mathfrak{h}$ to a black-box element $\lceil \mathfrak{h} \rfloor$ of R.
- Solve the DLP $(\lceil \mathfrak{h} \rfloor, \lceil \mathfrak{a} \rfloor)$ in the hidden version of R^* using Pohlig–Hellman. We get $k \in \mathbb{Z}$ such that $g^{\mathfrak{a}} = g^{\mathfrak{h}^k}$.
- Simply compute $\mathfrak{a}$ as the power $\mathfrak{h}^k \in R^*$.

Observation:
There is nothing special about using R^* in the exponents;
in principle anything expressible as field operations works.

Auxiliary groups

Observation:
There is nothing special about using R^* in the exponents;
in principle anything expressible as field operations works.

This is known as an auxiliary group:
A smooth-order algebraic group over the black-box field.

Slightly late 'about me' slide

I played too many hacking competitions in $[2013; +\infty)$.

Example auxiliary group: $\mathbb{F}_{p^2}^*$

A challenge I made for a CTF last year:

Solve a DLP $(g, g^{\mathfrak{a}})$ in a black-box group of order $p = 2^{48} - 5297$ using at most 2^{14} queries to `mul`, `inv`, and `dhp`.

Example auxiliary group: $\mathbb{F}_{p^2}^*$

A challenge I made for a CTF last year:

> Solve a DLP $(g, g^{\mathfrak{a}})$ in a black-box group of order $p = 2^{48} - 5297$ using at most 2^{14} queries to `mul`, `inv`, and `dhp`.

```
p = 2**48 - 5297            # |G| = p
mul = lambda x,y: (x+y)%p   # g^x * g^y = g^(x+y)
inv = lambda x: (-x)%p      # (g^x)^-1  = g^(-x)
dhp = lambda x,y: x*y%p     # enjoy your oracle!

aes = AES.new(os.urandom(16), AES.MODE_ECB)
enc = lambda x: aes.encrypt(x.to_bytes(16, 'big')).hex()
dec = lambda y: int.from_bytes(aes.decrypt(bytes.fromhex(y)), 'big')

g, a = 1, random.randrange(p)
print(enc(g), enc(y))

for _ in range(2**14):
    q = input().strip().split()
    if q[0] == 'mul': print(enc(mul(dec(q[1]), dec(q[2]))))
    if q[0] == 'inv': print(enc(inv(dec(q[1]))))
    if q[0] == 'dhp': print(enc(dhp(dec(q[1]), dec(q[2]))))

if int(input()) % p == a: print(open('flag.txt').read())
```

Example auxiliary group: $\mathbb{F}_{p^2}^*$

A challenge I made for a CTF last year:

Solve a DLP $(g, g^{\mathfrak{a}})$ in a black-box group of order $p = 2^{48} - 5297$ using at most 2^{14} queries to `mul`, `inv`, and `dhp`.

Intended solution: next slide.

Example auxiliary group: $\mathbb{F}_{p^2}^*$

A challenge I made for a CTF last year:

> Solve a DLP $(g, g^{\mathfrak{a}})$ in a black-box group of order $p = 2^{48} - 5297$ using at most 2^{14} queries to `mul`, `inv`, and `dhp`.

Intended solution: next slide.

Better solution: [https://sasdf.cf/ctf/writeup/2018/hxp/crypto/blinder_v2/]

- Notice $p + 1 = 2^4 \cdot 3 \cdot 5 \cdot 59 \cdot 281 \cdot 3037 \cdot 23293$ is smooth.

Example auxiliary group: $\mathbb{F}_{p^2}^*$

A challenge I made for a CTF last year:

Solve a DLP $(g, g^{\mathfrak{a}})$ in a black-box group of order $p = 2^{48} - 5297$ using at most 2^{14} queries to `mul`, `inv`, and `dhp`.

Intended solution: next slide.

Better solution: [https://sasdf.cf/ctf/writeup/2018/hxp/crypto/blinder_v2/]

- Notice $p + 1 = 2^4 \cdot 3 \cdot 5 \cdot 59 \cdot 281 \cdot 3037 \cdot 23293$ is smooth.
- Write $\mathbb{F}_{p^2} = \{x + \mathrm{i}y : x, y \in \mathbb{F}_p\}$ and fix a generator $\mathfrak{h}$ of $\mathbb{F}_{p^2}^*$.

Example auxiliary group: $\mathbb{F}_{p^2}^*$

A challenge I made for a CTF last year:

> Solve a DLP $(g, g^{\mathfrak{a}})$ in a black-box group of order $p = 2^{48} - 5297$ using at most 2^{14} queries to `mul`, `inv`, and `dhp`.

Intended solution: next slide.

Better solution: [https://sasdf.cf/ctf/writeup/2018/hxp/crypto/blinder_v2/]

- Notice $p + 1 = 2^4 \cdot 3 \cdot 5 \cdot 59 \cdot 281 \cdot 3037 \cdot 23293$ is smooth.
- Write $\mathbb{F}_{p^2} = \{x + \mathrm{i}y : x, y \in \mathbb{F}_p\}$ and fix a generator $\mathfrak{h}$ of $\mathbb{F}_{p^2}^*$.
- Recover $k = \log_{\lceil \mathfrak{h} \rfloor}(\lceil \mathfrak{a} + \mathrm{i} \rfloor) \bmod (p+1)$ using the oracle. Thus $\mathfrak{h}^k = (\mathfrak{a} + \mathrm{i}) \cdot \mathfrak{h}^{r(p+1)}$ for some $r \in \mathbb{Z}$.

Example auxiliary group: $\mathbb{F}_{p^2}^*$

A challenge I made for a CTF last year:

> Solve a DLP $(g, g^{\mathfrak{a}})$ in a black-box group of order $p = 2^{48} - 5297$ using at most 2^{14} queries to `mul`, `inv`, and `dhp`.

Intended solution: next slide.

Better solution: [https://sasdf.cf/ctf/writeup/2018/hxp/crypto/blinder_v2/]

- Notice $p + 1 = 2^4 \cdot 3 \cdot 5 \cdot 59 \cdot 281 \cdot 3037 \cdot 23293$ is smooth.
- Write $\mathbb{F}_{p^2} = \{x + \mathrm{i}y : x, y \in \mathbb{F}_p\}$ and fix a generator $\mathfrak{h}$ of $\mathbb{F}_{p^2}^*$.
- Recover $k = \log_{\lceil \mathfrak{h} \rfloor}(\lceil \mathfrak{a} + \mathrm{i} \rfloor) \bmod (p+1)$ using the oracle. Thus $\mathfrak{h}^k = (\mathfrak{a} + \mathrm{i}) \cdot \mathfrak{h}^{r(p+1)}$ for some $r \in \mathbb{Z}$.
- Observe that $\mathfrak{h}^{p+1} \in \mathbb{F}_p^*$, so $\mathfrak{h}^k = c\mathfrak{a} + c\mathrm{i}$ for some $c \in \mathbb{F}_p^*$. $\implies$ Divide $\mathfrak{h}^k$ by its i-coefficient to obtain $\mathfrak{a} + \mathrm{i}$, hence $\mathfrak{a}$!

Example auxiliary group: $\mathbb{F}_{p^2}^*$

A challenge I made for a CTF last year:

Solve a DLP $(g, g^{\mathfrak{a}})$ in a black-box group of order $p = 2^{48} - 5297$ using at most 2^{14} queries to `mul`, `inv`, and `dhp`.

Intended solution: next slide.

Better solution: [https://sasdf.cf/ctf/writeup/2018/hxp/crypto/blinder_v2/]

- Notice $p + 1 = 2^4 \cdot 3 \cdot 5 \cdot 59 \cdot 281 \cdot 3037 \cdot 23293$ is smooth.
- Write $\mathbb{F}_{p^2} = \{x + \mathrm{i}y : x, y \in \mathbb{F}_p\}$ and fix a generator $\mathfrak{h}$ of $\mathbb{F}_{p^2}^*$.
- Recover $k = \log_{\lceil \mathfrak{h} \rfloor}(\lceil \mathfrak{a} + \mathrm{i} \rfloor) \bmod (p+1)$ using the oracle. Thus $\mathfrak{h}^k = (\mathfrak{a} + \mathrm{i}) \cdot \mathfrak{h}^{r(p+1)}$ for some $r \in \mathbb{Z}$.
- Observe that $\mathfrak{h}^{p+1} \in \mathbb{F}_p^*$, so $\mathfrak{h}^k = c\mathfrak{a} + c\mathrm{i}$ for some $c \in \mathbb{F}_p^*$. $\Longrightarrow$ Divide $\mathfrak{h}^k$ by its i-coefficient to obtain $\mathfrak{a} + \mathrm{i}$, hence $\mathfrak{a}$!

(This uses only $\sim$4500 queries. Intended solution $\sim$3 times as many.)

Second result: Maurer (1994)

Let G be of prime order p, write $R = \mathbb{F}_p$,
and suppose $E\colon y^2 = x^3 + Ax^2 + x \ /\mathbb{F}_p$ has smooth order.
Then CDH is polynomial-time equivalent to DLP in G.

Second result: Maurer (1994)

Let G be of prime order p, write $R = \mathbb{F}_p$,
and suppose $E\colon y^2 = x^3 + Ax^2 + x\ /\mathbb{F}_p$ has smooth order.
Then CDH is polynomial-time equivalent to DLP in G.

Proof:

- Find a generator point G on E.
- Hope that $\mathfrak{a}$ is an x-coordinate on the curve ($\Pr \approx 1/2$).
  Compute (black-box) the corresponding y-coordinate $\lceil \mathfrak{y} \rfloor$,
  giving a black-box elliptic-curve point $\lceil P \rfloor = (\lceil \mathfrak{a} \rfloor, \lceil \mathfrak{y} \rfloor)$.
  (If $\lceil \mathfrak{y} \rfloor^2 \neq \lceil \mathfrak{a} \rfloor^3 + \lceil A \rfloor \lceil \mathfrak{a} \rfloor^2 + \lceil \mathfrak{a} \rfloor$, then randomize $\lceil \mathfrak{a} \rfloor$ as $\lceil \mathfrak{a}' \rfloor = \lceil \mathfrak{a} \rfloor + \lceil \delta \rfloor$ and retry.)
- Solve the (black-box) DLP $(\lceil G \rfloor, \lceil P \rfloor)$ via Pohlig–Hellman.
  We get $k \in \mathbb{Z}$ such that $(\mathfrak{a}, \mathfrak{y}) = [k]G$.
- Simply compute $\mathfrak{a}$ as the x-coordinate of $[k]G$.

Second result: Maurer (1994)

Let G be of prime order p, write $R = \mathbb{F}_p$,
and suppose $E\colon y^2 = x^3 + Ax^2 + x \ /\mathbb{F}_p$ has smooth order.
Then CDH is polynomial-time equivalent to DLP in G.

Are there always such E?

Unknown in general, but likely.

People have constructed some for 'common' groups G.

$\implies$ For all practical purposes, DLP is equivalent to CDH.

Second result: Maurer (1994)

Let G be of prime order p, write $R = \mathbb{F}_p$,
and suppose $E\colon y^2 = x^3 + Ax^2 + x \ /\mathbb{F}_p$ has smooth order.
Then CDH is polynomial-time equivalent to DLP in G.

Are there always such E?

Unknown in general, but likely.

People have constructed some for 'common' groups G.

$\Longrightarrow$ For all practical purposes, DLP is equivalent to CDH.

...and they lived happily ever after**??**

Shor's algorithm (1994)

Shor's algorithm breaks all group-based DH instantiations.

Shor's algorithm (1994)

...is a quantum algorithm for period finding.

Shor's algorithm (1994)

...is a quantum algorithm for period finding.

Let S be some finite set and

$$f\colon\ \mathbb{Z}^n \longrightarrow S$$

a map with an unknown period lattice $\Lambda \subseteq \mathbb{Z}^n$, such that

$$f(v+\tau) = f(v)$$

if and only if $\tau \in \Lambda$.

Shor's algorithm (1994)

...is a quantum algorithm for period finding.

Let S be some finite set and

$$f\colon\ \mathbb{Z}^n \longrightarrow S$$

a map with an unknown period lattice $\Lambda \subseteq \mathbb{Z}^n$, such that

$$f(v+\tau) = f(v)$$

if and only if $\tau \in \Lambda$.

Given such f and some size constraints on Λ,
Shor's algorithm recovers a basis of Λ in polynomial time.

Shor's algorithm (1994)

...is a quantum algorithm for period finding.

Application:

For a DLP instance $(g, h = g^{\mathfrak{a}})$ in a cyclic group G of order q, the (publicly computable) function

$$\begin{aligned} f\colon\ \mathbb{Z}^2 &\longrightarrow G \\ (x, y) &\longmapsto g^x \cdot h^y \end{aligned}$$

has period $\Lambda = \langle (\mathfrak{a}, -1), (q, 0) \rangle \subseteq \mathbb{Z}^2$, which Shor can recover.

And now...
For something totally different.

Diffie–Hellman from group actions (2006)

Let G be a group, X a set. A group action of G on X is a map

$$*\colon\; G \times X \longrightarrow X$$

such that $\mathrm{id} * x = x$ and $(\mathfrak{g} \cdot \mathfrak{h}) * x = \mathfrak{g} * (\mathfrak{h} * x)$.

Diffie–Hellman from group actions (2006)

Let G be a group, X a set. A group action of G on X is a map

$$*\colon\ G \times X \longrightarrow X$$

such that $\mathrm{id} * x = x$ and $(\mathfrak{g} \cdot \mathfrak{h}) * x = \mathfrak{g} * (\mathfrak{h} * x)$.

This suggests an obvious Diffie–Hellman scheme:
Let G be finite and commutative and fix $x \in X$.

- Private keys: group elements $\mathfrak{a}, \mathfrak{b} \in G$.
- Public keys: the elements $\mathfrak{a} * x$, $\mathfrak{b} * x \in X$.
- Shared secret: the element $\mathfrak{a} * (\mathfrak{b} * x) = \mathfrak{b} * (\mathfrak{a} * x)$.

Diffie–Hellman from group actions (2006)

Let G be a group, X a set. A group action of G on X is a map

$$*\colon\ G \times X \longrightarrow X$$

such that $\mathrm{id} * x = x$ and $(\mathfrak{g} \cdot \mathfrak{h}) * x = \mathfrak{g} * (\mathfrak{h} * x)$.

This suggests an obvious Diffie–Hellman scheme:
Let G be finite and commutative and fix $x \in X$.

- Private keys: group elements $\mathfrak{a}, \mathfrak{b} \in G$.
- Public keys: the elements $\mathfrak{a} * x$, $\mathfrak{b} * x \in X$.
- Shared secret: the element $\mathfrak{a} * (\mathfrak{b} * x) = \mathfrak{b} * (\mathfrak{a} * x)$.

This is not in general broken by Shor!

Diffie–Hellman from group actions (2006)

Let G be a group, X a set. A group action of G on X is a map

$$*\colon\ G \times X \longrightarrow X$$

such that $\mathrm{id} * x = x$ and $(\mathfrak{g} \cdot \mathfrak{h}) * x = \mathfrak{g} * (\mathfrak{h} * x)$.

This suggests an obvious Diffie–Hellman scheme:
Let G be finite and commutative and fix $x \in X$.

- Private keys: group elements $\mathfrak{a}, \mathfrak{b} \in G$.
- Public keys: the elements $\mathfrak{a} * x,\ \mathfrak{b} * x \in X$.
- Shared secret: the element $\mathfrak{a} * (\mathfrak{b} * x) = \mathfrak{b} * (\mathfrak{a} * x)$.

This is not in general broken by Shor!

Example: CSIDH [ˈsiːˌsaɪd] (2018) [joint w/ Castryck, Lange, Martindale, Renes]

The implicit group

Just like before, we get an implicit structure on the public keys.

The implicit group

Just like before, we get an implicit structure on the public keys.

However, crucially, the operation $g^{\mathfrak{a}} \cdot g^{\mathfrak{b}} = g^{\mathfrak{a}+\mathfrak{b}}$ is lost.

$\Longrightarrow$ We only get a black-box group rather than a ring or field.

The implicit group

Just like before, we get an implicit structure on the public keys.

However, crucially, the operation $g^{\mathfrak{a}} \cdot g^{\mathfrak{b}} = g^{\mathfrak{a}+\mathfrak{b}}$ is lost.

$\Longrightarrow$ We only get a black-box group rather than a ring or field.

Anyone can...

- encode elements $\mathfrak{a}$: compute $\lceil \mathfrak{a} \rfloor = \mathfrak{a} * x$.
- translate by cleartext elements: $\mathfrak{a} * \lceil \mathfrak{b} \rfloor = \lceil \mathfrak{a} \cdot \mathfrak{b} \rfloor$.

The implicit group

Just like before, we get an implicit structure on the public keys.

However, crucially, the operation $g^{\mathfrak{a}} \cdot g^{\mathfrak{b}} = g^{\mathfrak{a}+\mathfrak{b}}$ is lost.

$\Longrightarrow$ We only get a black-box group rather than a ring or field.

Anyone can...

- encode elements $\mathfrak{a}$: compute $\lceil \mathfrak{a} \rfloor = \mathfrak{a} * x$.
- translate by cleartext elements: $\mathfrak{a} * \lceil \mathfrak{b} \rfloor = \lceil \mathfrak{a} \cdot \mathfrak{b} \rfloor$.

Anyone who can solve CDH can...

- compose: $\lceil \mathfrak{a} \rfloor \cdot \lceil \mathfrak{b} \rfloor = \mathit{shared\_secret}(x, \mathfrak{a} * x, \mathfrak{b} * x)$.

The implicit group

Just like before, we get an implicit structure on the public keys.

However, crucially, the operation $g^{\mathfrak{a}} \cdot g^{\mathfrak{b}} = g^{\mathfrak{a}+\mathfrak{b}}$ is lost.

$\Longrightarrow$ We only get a black-box group rather than a ring or field.

Anyone can...

- encode elements $\mathfrak{a}$: compute $\lceil \mathfrak{a} \rfloor = \mathfrak{a} * x$.
- translate by cleartext elements: $\mathfrak{a} * \lceil \mathfrak{b} \rfloor = \lceil \mathfrak{a} \cdot \mathfrak{b} \rfloor$.

Anyone who can solve CDH can...

- compose: $\lceil \mathfrak{a} \rfloor \cdot \lceil \mathfrak{b} \rfloor = \mathit{shared\_secret}(x, \mathfrak{a} * x, \mathfrak{b} * x)$.
- exponentiate: square-and-multiply using $\circlearrowleft$.

The implicit group

Just like before, we get an implicit structure on the public keys.

However, crucially, the operation $g^{\mathfrak{a}} \cdot g^{\mathfrak{b}} = g^{\mathfrak{a}+\mathfrak{b}}$ is lost.

$\Longrightarrow$ We only get a black-box group rather than a ring or field.

Anyone can...

- encode elements $\mathfrak{a}$: compute $\lceil \mathfrak{a} \rfloor = \mathfrak{a} * x$.
- translate by cleartext elements: $\mathfrak{a} * \lceil \mathfrak{b} \rfloor = \lceil \mathfrak{a} \cdot \mathfrak{b} \rfloor$.

Anyone who can solve CDH can...

- compose: $\lceil \mathfrak{a} \rfloor \cdot \lceil \mathfrak{b} \rfloor = \mathit{shared\_secret}(x, \mathfrak{a} * x, \mathfrak{b} * x)$.
- exponentiate: square-and-multiply using .
- invert: $\lceil \mathfrak{a}^{-1} \rfloor = \lceil \mathfrak{a} \rfloor^{|G|-1}$ using .

Our result (2018) [joint w/ Galbraith, Smith, Vercauteren]

Theorem. There is a polynomial-time quantum equivalence between the CDH and DLP problems for group actions.

Our result (2018) [joint w/ Galbraith, Smith, Vercauteren]

Theorem. There is a polynomial-time quantum equivalence between the CDH and DLP problems for group actions.

Proof:

- Compute a set of generators $\mathfrak{g}_1, \ldots, \mathfrak{g}_r \in G$.
- Apply Shor's algorithm to the map

$$\begin{aligned} f\colon \quad \mathbb{Z}^r \times \mathbb{Z} &\longrightarrow X \\ (x_1, \ldots, x_r, y) &\longmapsto (\mathfrak{g}_1^{x_1} \cdots \mathfrak{g}_r^{x_r}) * \lceil \mathfrak{a} \rfloor^y . \end{aligned}$$

- Any period vector of the form $(x_1, \ldots, x_r, 1)$ yields the desired element $\mathfrak{a} = \mathfrak{g}_1^{-x_1} \cdots \mathfrak{g}_r^{-x_r}$.

Work in "progress"

- Can we get similar results in the group-action setting if the CDH oracle $(x, \mathfrak{a} * x, \mathfrak{b} * x) \mapsto \mathfrak{a}\mathfrak{b} * x$ is unreliable?

  Classical case: Yes, by repeatedly blinding the inputs, unblinding the outputs, and using majority vote.

  Here: Exponentially many queries in superposition; do we need *all* of them to be correct?

Work in "progress"

- Can we get similar results in the group-action setting if the CDH oracle $(x, \mathfrak{a} * x, \mathfrak{b} * x) \mapsto \mathfrak{a}\mathfrak{b} * x$ is unreliable?

Likely. But we haven't worked this out yet.

And it might turn out to be impossible.

So probably it's best if you forget about it.

At least for the time being.

Until we've worked it out.

Hopefully soon.

Work in "progress"

- Can we get similar results in the group-action setting if the CDH oracle $(x, \mathfrak{a} * x, \mathfrak{b} * x) \mapsto \mathfrak{ab} * x$ is unreliable?

Likely. But we haven't worked this out yet.

And it might turn out to be impossible.

So probably it's best if you forget about it.

At least for the time being.

Until we've worked it out.

Hopefully soon.

Thank you!