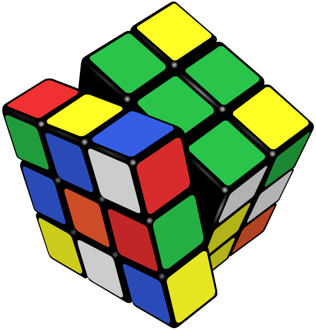
Cryptography in Cyclic Groups



Overview

Groups

- A **group** is a set $\mathbb{G}$ along with a binary operation
  - Additive notation of multiplicative notation
- There is a **neutral element** (denoted by 0 or 1)
- Each group element has an **inverse** (denoted by $-x$ or x^{-1})

In cryptology

Two kinds of groups are widely used:

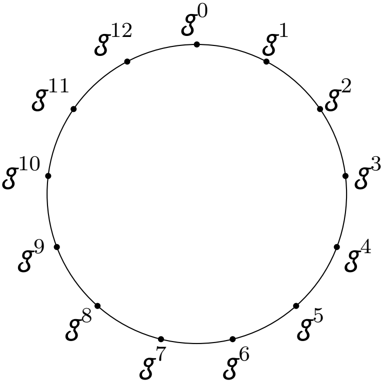
1. Invertible integers modulo N, in particular when N is prime
2. Points on an elliptic curve ($y^2 = x^3 + ax + b$)

- If $(G, \times)$ is a group, $H \subseteq G$ and $(H, \times)$ is also a group
  - Then H is a **subgroup** of G.
- If G is a finite group, then $|G|$ is the **order** of G

Cyclic Groups

Let $\mathbb{G}$ be a finite group of order N and $g \in \mathbb{G}$

- The **cyclic group generated by** g is $\langle g \rangle = \{g^i \mid i \in \mathbb{Z}\}$
- This is obviously a subgroup of $\mathbb{G}$...
- ... therefore $\langle g \rangle$ is finite of order $q \leq N$
- q is the **order** of g ($=$ of the cyclic subgroup generated by g)



Lemma

Let q denote the order of $\langle g \rangle$. If $i > 0$ and $g^i = 1$, then $q \leq i$.

Proof.

- Let $u \in \mathbb{Z}$
- Euclidean division of u by i: $u = ki + r$ with $r < i$
- $g^u = g^{ki+r} = \left(g^i\right)^k \cdot g^r = 1^k \cdot g^r = g^r$
- g^u can take at most i distinct values, therefore $q \leq i$.

$\square$

Proposition

Let q denote the order of $\langle g \rangle$. Then q is the smallest $i > 0$ s.t. $g^i = 1$.

Proof.

1. For $1 \leq i < q, g^i \neq 1$
   - Suppose not: $g^i = 1$ with $i < q$
   - Previous lemma yields $q \leq i < q$
2. For $0 \leq i < q$, the g^i are all different
   - Suppose not: $g^i = g^j$ with $i < j < q$
   - Therefore $g^{j-i} = 1$ and $1 \leq j - i < q$
3. $g^q = g^k$ for some $0 \leq k < q$
   - Suppose not: then $q + 1$ elements of $\langle g \rangle$ are distinct
4. $k = 0$
   - $g^{q-k} = 1$, and the previous lemma yields $q \leq q - k$

□

Proposition

Let q denote the order of $\langle g \rangle$. Then:

$$g^u = g^v \iff u \equiv v \bmod q$$

Proof.

Suppose $u \geq v$; Euclidean division by q: $u - v = qi + r$ ($r < q$)

$$\begin{aligned}
g^u = g^v &\Leftrightarrow g^{u-v} = 1 \\
&\Leftrightarrow g^{iq+r} = 1 \\
&\Leftrightarrow (g^q)^i \cdot g^r = 1 \\
&\Leftrightarrow 1^i \cdot g^r = 1 \\
&\Leftrightarrow g^r = 1 \\
&\Leftrightarrow r = 0 && \text{(by previous proposition, } r < q\text{)} \\
&\Leftrightarrow u - v = iq \\
&\Leftrightarrow u \equiv v \mod q
\end{aligned}$$

In a cyclic group of order q,
exponents are always “mod q”

Multiplicative Groups of Integer Modulo p

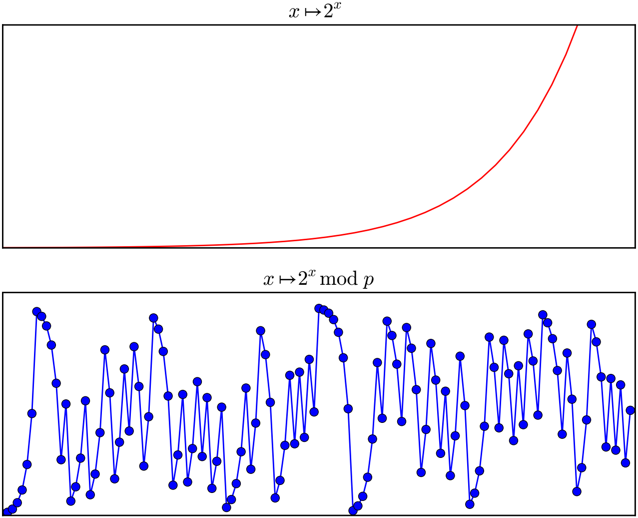
- $\mathbb{Z}_p^\times = \{1, 2, \ldots, p-1\}$ = invertible integers mod p
- Order $p - 1$

Main interest

Discrete logarithm is (presumably) hard

- Given $g^x \bmod p$, no efficient algorithm to find x

Exponentiation Modulo p Is Not Easy to Invert

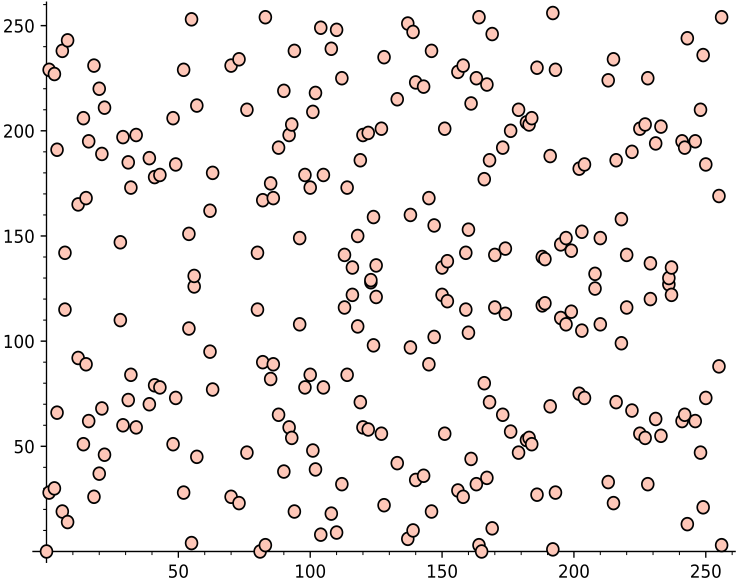


Curve25519

$y^2 = x^3 + 486662x^2 + x$

Curve25519

$y^2 = x^3 + 486662x^2 + x \mod 2^{255} - 19$



Discrete Logarithm in $\mathbb{Z}_p^\times$

Given a **generator** g and a **target** $h = g^x$, find x

Observations

- Let q denote the order of g modulo p
  $=$ the order of g in $\mathbb{Z}_p^\times$
- x is defined "modulo q"
  - Choosing x uniformly in $[0; q)$ is sufficient
- Simple approach: **exhaustive search**
  - For $i = 0, 1, 2, \ldots, q-1$: if $h = g^i$ then return i
  - Complexity: q multiplications by g and equality tests

$\leadsto$ Need generators of **large order**

Discrete Logarithm in $\mathbb{Z}_p^\times$
Continued

Given a *generator* g of order q and a *target* $h = g^x$, find x

Best algorithms

- Number Field Sieve
  - Complexity $\mathcal{O}\left(\exp((1.92 + o(1))(\log p)^{1/3}(\log\log p)^{2/3})\right)$
  - (Depends only of p)
  - Current record: 795-bit p (2020). 3200 CPU-year.
  - Security $\rightsquigarrow$ large p (2000-3000 bits)

- Pollard rho
  - Complexity $\mathcal{O}\left(\sqrt{q}\right)$
  - Current record: 112-bit q (2012) cluster of Playstation 3
  - Security $\rightsquigarrow$ large q (256 bits)

- Pohlig-Hellman
  - If $q = uv$, then project into subgroups of order u,v
  - Security $\rightsquigarrow$ q with large prime factor (256 bits)

Questions

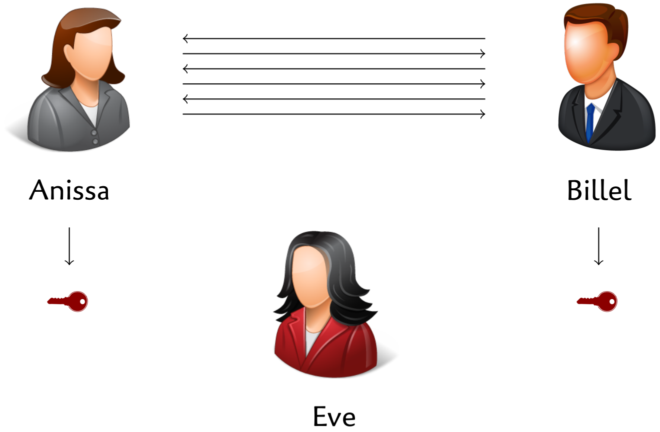
1. How to find g of order q s.t. q has a prime factor $\geq 2^{256}$?
2. How to determine the order of g?
3. Do random g have large order modulo p?
4. What is the largest possible order of g modulo p?

Discrete Logarithm in $\mathbb{Z}_p^\times$

Questions

1. How to find g of order q s.t. q has a prime factor $\geq 2^{256}$?
   - **EASY** (if one can choose p)
2. How to determine the order of g?
   - **HARD** in general
3. Do random g have large order modulo p?
   - **YES** (mostly)
4. What is the largest possible order of g modulo p?
   - $p-1$

Key Exchange



Diffie-Hellman Key Exchange
1976

$(\mathbb{G}, \cdot)$ a finite cyclic group; $\langle g \rangle = \mathbb{G}$

Anissa

Billel

Eve

Diffie-Hellman Key Exchange
1976

$(\mathbb{G}, \cdot)$ a finite cyclic group; $\langle g \rangle = \mathbb{G}$

Anissa

$$\xrightarrow{\qquad y_a = g^a \qquad}$$

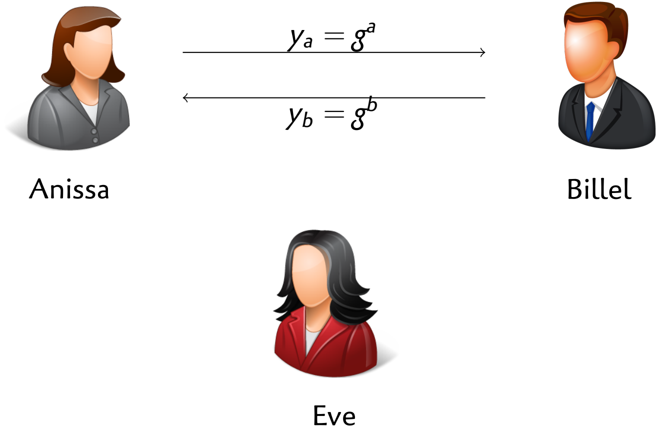
Billel

Eve

Diffie-Hellman Key Exchange

1976

$(\mathbb{G}, \cdot)$ a finite cyclic group; $\langle g \rangle = \mathbb{G}$



Diffie-Hellman Key Exchange

1976

$(\mathbb{G}, \cdot)$ a finite cyclic group; $\langle g \rangle = \mathbb{G}$

$$\xrightarrow{\quad y_a = g^a \quad}$$

$$\xleftarrow[\quad y_b = g^b \quad]{}$$

Anissa

$\downarrow$

$K_a = {y_b}^a$

Billel

$\downarrow$

$K_b = {y_a}^b$

Eve

Diffie-Hellman Key Exchange
1976

$(\mathbb{G}, \cdot)$ a finite cyclic group; $\langle g \rangle = \mathbb{G}$

$$\xrightarrow{y_a = g^a}$$

$$\xleftarrow[y_b = g^b]{}$$

Anissa

Billel

$$\downarrow$$

$$K_a = {y_b}^a$$

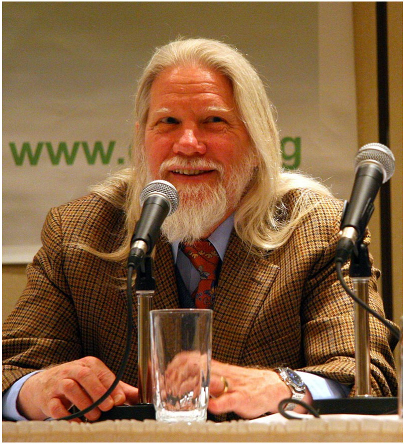
$$\downarrow$$

$$K_b = {y_a}^b$$

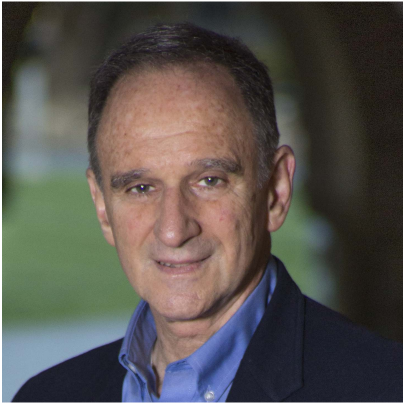
Eve

$$K_a = {y_b}^a = (g^b)^a = g^{ab} = (g^a)^b = {y_a}^b = K_b$$

Diffie-Hellman Key Exchange
1976



Whitfield Diffie
(1944–)



Martin E. Hellman
(1945–)

Diffie-Hellman Key Exchange: Security

Eve knows:

- g
- $y_a = g^a$
- $y_b = g^b$

and should have no information on $K = g^{ab}$

- If finding a from y_a is easy then the DH key exchange is not secure.

Elgamal Encryption
1984

- Non-interactive version of Diffie-Hellman key-exchange
- Group $\mathbb{G}$, cyclic subgroup $\langle g \rangle$ of order q

Key Generation

- Choose random integer $0 \leq x < q$
- Compute $h \leftarrow g^x$

Public key $= \mathbb{G}, g, h$

Secret key $= x$

Elgamal Encryption
1984

Encryption

- Message space = $\mathbb{G}$
- Choose random integer $0 \leq r < q$
- Ciphertext: $c \leftarrow (g^r, h^r \cdot m)$

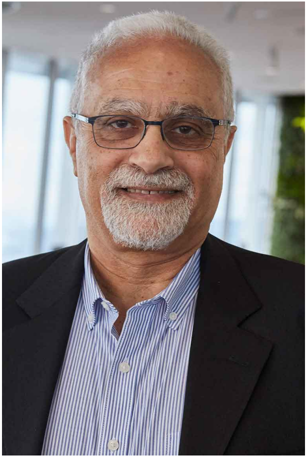
(non-deterministic)

Decryption

- Ciphertext $c = (a, b)$
- Output $(a^x)^{-1} \cdot b$

$h = g^x, a = g^r$ and $b = h^r \cdot m \longrightarrow (a^x)^{-1} \cdot b = g^{-rx} \cdot h^r \cdot m = m$

Elgamal Encryption
1984



ظاهر الجمل* (1955–)

* Taher Elgamal

Relevant Algorithmic Problems

DLOG Given g, g^x, find x

CDH Given g, g^x, g^y, find g^{xy} (Computational Diffie-Hellman)

DDH Compute $\mathcal{F}$ (Decisional Diffie-Hellman)

$$\mathcal{F}(g,h,u,v) = \begin{cases} 1 & \text{if } \exists x.\ u = g^x \text{ and } v = h^x \\ 0 & \text{otherwise} \end{cases}$$

Observations

- **DLOG** easy $\Longrightarrow$ **CDH** easy $\Longrightarrow$ **DDH** easy
- Elgamal key recovery $\Longleftrightarrow$ **DLOG**
  - Public key h^x / Secret key $= x$
- Elgamal **OW** $\Longleftrightarrow$ **CDH**
  - **CDH** easy $\Longrightarrow$ compute h^r from g, h, g^r
  - Elgamal not **OW** $\Longrightarrow$ set $h = g^x$, $m \leftarrow \mathcal{A}(g^y, \alpha)$, $\alpha \cdot m^{-1} = g^{xy}$

Relevant Algorithmic Problems

DLOG Given g, g^x, find x

CDH' Given g, h, g^x, find h^x (equivalent CDH variation)

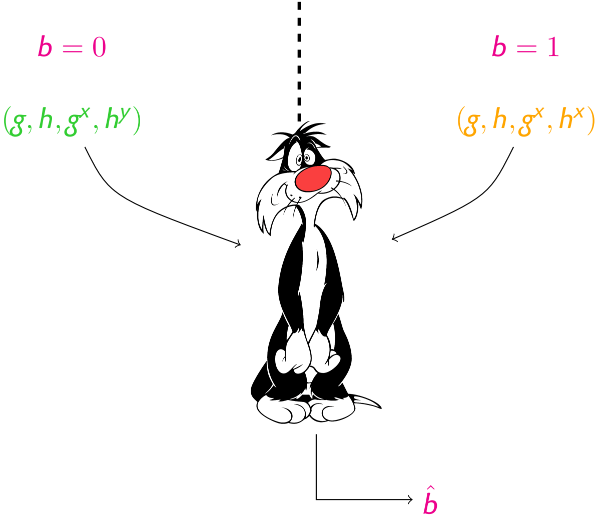
DDH Compute $\mathcal{F}$ (Decisional Diffie-Hellman)

$$\mathcal{F}(g, h, u, v) = \begin{cases} 1 & \text{if } \exists x.\ u = g^x \text{ and } v = h^x \\ 0 & \text{otherwise} \end{cases}$$

Observations

- **DLOG** easy $\Longrightarrow$ **CDH** easy $\Longrightarrow$ **DDH** easy
- Elgamal key recovery $\Longleftrightarrow$ **DLOG**
  - Public key h^x / Secret key $= x$
- Elgamal **OW** $\Longleftrightarrow$ **CDH**
  - **CDH** easy $\Longrightarrow$ compute h^r from g, h, g^r
  - Elgamal not **OW** $\Longrightarrow$ set $h = g^x$, $m \leftarrow \mathcal{A}(g^y, \alpha)$, $\alpha \cdot m^{-1} = g^{xy}$

DDH — Alternative Point of View



- Distinguisher must tell if he is in "world $b = 0$"...
- ... or in "world $b = 1$"

Compute $\mathcal{F}$ (Decisional Diffie-Hellman)

$$\mathcal{F}(g, h, u, v) = \begin{cases} 1 & \text{if } \exists x.\ u = g^x \text{ and } v = h^x \\ 0 & \text{otherwise} \end{cases}$$

Simple strategy to compute $\mathcal{F}$

- Just return a random bit! Correct with proba. 50%

Concept of **advantage**

$\Rightarrow$ Disqualify naive strategies

- Advantage of an algorithm $\mathcal{A}$:

$$\mathbf{Adv}_{DDH}(\mathcal{A}) = \left|\Pr(\mathcal{A} \to 1 \mid b = 1) - \Pr(\mathcal{A} \to 1 \mid b = 0)\right|$$

- Random guess / constant answer $\rightsquigarrow$ advantage 0
- Correct all the time $\rightsquigarrow$ advantage 1
- **DDH hard** $\iff$ efficient algo. have **negligible** advantage

DDH Can be Easier than CDH

Let g be a primitive root modulo p

- **DLOG** and **CDH** are (presumably) hard in $\mathbb{Z}_p^\times$
- **DDH** is easy in $\mathbb{Z}_p^\times$!!!
- Argument given around 1800

Leonhard Euler
1707–1783

Adrien-Marie Legendre
1752–1833

- Stay tuned for next lecture on **HARDCODE PREDICATES**!

Semantic Security of Elgamal (a.k.a. IND-CPA)

World b

$$\xleftarrow{\quad h = g^{x} \quad}$$

$$\xrightarrow{\quad m_0, m_1 \quad}$$

$$\xleftarrow{\quad (g^{r}, h^{r} \cdot m_b) \quad}$$

x

r

Adversary

Challenger

$\hookrightarrow \hat{b}$

$$\mathbf{Adv}_{IND}(\mathcal{A}) = \left|\Pr(\mathcal{A} \to 1 \mid b = 1) - \Pr(\mathcal{A} \to 1 \mid b = 0)\right|$$

(measures capacity of $\mathcal{A}$ to learn information from the ciphertext)

Semantic Security of Elgamal (a.k.a. IND-CPA)

Theorem

Elgamal is **IND-CPA** $\Longleftrightarrow$ **DDH** is hard

Proof.

1. Suppose **DDH** is easy
   - Build good IND-CPA adversary

$\square$

$$h = g^{x}$$
$$\longleftarrow$$
$$m_0 \neq 1, m_1 = 1$$
$$\longrightarrow$$
$$(g^{r}, h^{r} \cdot m_b)$$
$$\longleftarrow$$

x

r

Adversary $\mathcal{B}$ — Challenger

$$\longrightarrow \mathcal{A}(g, h, g^{r}, m_b \cdot h^{r})$$

- DDH easy:
  - $\exists$ efficient $\mathcal{A}$ that computes $\mathcal{F}$ correctly w/ high proba
- $m_0 \neq 1, m_1 = 1 \implies \mathcal{F}(g, h, g^{r}, h^{r} \cdot m_b) = b$
- $\mathcal{A}$ answers DDH correctly $\implies \mathcal{B}$ answers IND-CPA correctly
  - $\mathbf{Adv}_{IND}(\mathcal{B}) = \mathbf{Adv}_{DDH}(\mathcal{A})$

Semantic Security of Elgamal (a.k.a. IND-CPA)

Theorem

Elgamal is **IND-CPA** $\iff$ **DDH** is hard

Proof.

1. Suppose **DDH** is easy
   - Build good IND-CPA adversary
2. Suppose Elgamal is not **IND-CPA**
   - Build efficient DDH algorithm w/ non-negligible advantage

□

(g, h, u, v)

$\downarrow$

DDH

Adversary $\mathcal{B}$

$\downarrow$

b

(g, h, u, v)

$\downarrow$

DDH

IND-CPA

$$\xleftarrow{\text{pk}}$$

$$\xrightarrow{m_0,\, m_1}$$

$$\xleftarrow[\text{(“challenge”)}]{\text{ciphertext}}$$

Adversary $\mathcal{A}$ $\longrightarrow \hat{b}'$ - - - → Adversary $\mathcal{B}$ $\longrightarrow b$

- $\mathcal{B}$ uses $\mathcal{A}$ to solve DDH

IND-CPA

pk $\longleftarrow$

m_0, m_1 $\longrightarrow$

ciphertext $\longleftarrow$

(“challenge”)

Adversary $\mathcal{A}$

Challenger

$\hat{b}'$

- $\mathcal{B}$ uses $\mathcal{A}$ to solve DDH

$\Rightarrow$ Must **faithfully simulate** the challenger that $\mathcal{A}$ expects

- (otherwise knothing is known about the answers of $\mathcal{A}$)

$\boxed{\text{World } b = 0}$

(g, h, g^x, h^y)

$\downarrow$

IND-CPA

DDH

$\xleftarrow{\quad h \quad}$

$\xrightarrow{\quad m_0, m_1 \quad}$

$\xleftarrow{\quad (g^x, h^y \cdot m_{b'}) \quad}$

$b' \xleftarrow{\$} \{0, 1\}$

Adversary $\mathcal{A}$

Adversary $\mathcal{B}$

$\hat{b}' = \$\$\$$

$[b' = \hat{b}']$

- ciphertext = 2 **random** group elements
- no information about b' at all
- Proba. that $\mathcal{A}$ guesses $b' = 50\%$
- $\Pr(\mathcal{B} \to 1 \mid b = 0) = 0.5$

World $b = 1$

(g, h, g^x, h^x)

DDH

IND-CPA

$\xleftarrow{\quad h \quad}$

$\xrightarrow{\quad m_0, m_1 \quad}$

$\xleftarrow{\quad (g^x, h^x \cdot m_{b'}) \quad}$

$b' \xleftarrow{\$} \{0, 1\}$

Adversary $\mathcal{A}$ Adversary $\mathcal{B}$

$\hat{b}'$

$[b' = \hat{b}']$

- $\mathcal{B}$ faithfully simulates the Challenger
- Proba. that $\mathcal{A}$ guesses b' correctly?

$$\mathcal{A} \text{ guesses } b' \iff (\mathcal{A} \to 1 \wedge b' = 1) \vee (\mathcal{A} \to 0 \wedge b' = 0)$$

$$\begin{aligned}
\Pr(\mathcal{A} \text{ guesses } b') &= \Pr(\mathcal{A} \to 1 \wedge b' = 1) + \Pr(\mathcal{A} \to 0 \wedge b' = 0) \\
&= \frac{1}{2}\Pr(\mathcal{A} \to 1 \mid b' = 1) + \frac{1}{2}\Pr(\mathcal{A} \to 0 \mid b' = 0)
\end{aligned}$$

Recall the definition:

$$\begin{aligned}
\mathbf{Adv}_{IND}(\mathcal{A}) &= \left|\Pr(\mathcal{A} \to 1 \mid b' = 1) - \Pr(\mathcal{A} \to 1 \mid b' = 0)\right| \\
&= \left|\Pr(\mathcal{A} \to 1 \mid b' = 1) - 1 + \Pr(\mathcal{A} \to 0 \mid b' = 0)\right| \\
&= \left|2 \cdot \Pr(\mathcal{A} \text{ guesses } b') - 1\right|
\end{aligned}$$

Finally:

$$\Pr(\mathcal{A} \text{ guesses } b') = \frac{1}{2}\mathbf{Adv}_{IND}(\mathcal{A}) + \frac{1}{2}$$

$\boxed{\text{World } b = 1}$

(g, h, g^x, h^x)

DDH

IND-CPA

$\xleftarrow{\quad h \quad}$

$\xrightarrow{\quad m_0, m_1 \quad}$

$\xleftarrow{\quad (g^x, h^x \cdot m_{b'}) \quad}$

$b' \xleftarrow{\$} \{0, 1\}$

Adversary $\mathcal{A}$

Adversary $\mathcal{B}$

$\hat{b}'$

$[b' = \hat{b}']$

- $\mathcal{B}$ faithfully simulates the Challenger
- Proba. that $\mathcal{A}$ guesses b' correctly?
- $\Pr(\mathcal{B} \to 1 \mid b = 1) = 0.5 + 0.5\mathbf{Adv}_{IND}(\mathcal{A})$

$\boxed{\text{World } b = 1}$

(g, h, g^x, h^x)

DDH

IND-CPA

$\xleftarrow{\quad h \quad}$

$\xrightarrow{\quad m_0, m_1 \quad}$

$\xleftarrow{\quad (g^x, h^x \cdot m_{b'}) \quad}$

$b' \xleftarrow{\$} \{0, 1\}$

Adversary $\mathcal{A}$

Adversary $\mathcal{B}$

$\hat{b}'$

$[b' = \hat{b}']$

- $\mathcal{B}$ faithfully simulates the Challenger
- Proba. that $\mathcal{A}$ guesses b' correctly?
- $\Pr(\mathcal{B} \to 1 \mid b = 1) = 0.5 + 0.5\mathbf{Adv}_{IND}(\mathcal{A})$
- $\mathbf{Adv}_{DDH}(\mathcal{B}) = 0.5\mathbf{Adv}_{IND}(\mathcal{A})$

Semantic Security of Elgamal (a.k.a. IND-CPA)

Theorem

Elgamal is **IND-CPA** $\iff$ **DDH** is hard

Proof.

1. Suppose **DDH** is easy
   - Build good IND-CPA adversary
2. Suppose Elgamal is not **IND-CPA**
   - Build efficient DDH algorithm w/ non-negligible advantage

$\square$

Joseph-Louis Lagrange
(1736–1813)

Theorem (Lagrange)

Let G be a finite group and $H \subseteq G$ a subgroup of G.
Then $|H|$ divides $|G|$.

Proof.

- Let $x, y \in G$
- Say that $x \sim y$ iff $\exists h \in H$ (the subgroup) such that $x = yh$
- $\sim$ is an equivalence relation (easy)
- The equivalence class of x is xH
- xH has cardinality $|H|$
  - Multiplication by x is a bijection in G
- Write $[G : H]$ the number of equivalence classes
  - Also known as the "*index* of H in G"
- The equivalence classes form a partition of G
- Therefore $|G| = [G : H] \times |H|$

$\square$

Interesting Consequence

Corollary

Let G be a group and $x \in G$. The order of x divides the order of G.

Proof.

$\langle x \rangle$ is a subgroup of G. Apply Lagrange's theorem. $\square$

Generators in $\mathbb{Z}_p^\times$

Let q denote the order of g modulo p

- $\mathbb{Z}_p^\times$ has order $p-1$
  - Notice that $p-1$ is even
  - $\{-1, 1\}$ is indeed a subgroup of order 2
- Therefore (Lagrange's theorem) q divides $p-1$
  - $\rightsquigarrow$ *Considerably restricts* the possible values of q
- q has a large prime factor $\Rightarrow p-1$ has a large prime factor
- $\mathbb{Z}_p^\times$ contains elements of order $p-1$
  - *Non-trivial theorem* (no proof given here)
  - This means that $\mathbb{Z}_p^\times$ is cyclic
  - An element of order $p-1$, is called a **primitive root** mod p

Checking the Order of a Generator

Problem

- Someone "promises" you that g has order q modulo p
- Can you verify that it is true?

Validation?

- Check that q divides $p-1$
- Check that $g \neq 1$
- Check that $g^q = 1$ (necessary, **not sufficient**)
  - This proves that the actual order of g divides q
  - It could be smaller than q
- Special case: the previous test is sufficient if q **is prime**,

Checking the Order of a Generator

Problem

- Someone "promises" you that g has order q modulo p
- q is **not prime** (relevant case: primitive roots)

Validation?

- Let ℓ denote the actual order of g
- Check that $g^q = 1$ (necessary, **not sufficient**)
  - This proves that ℓ divides q
  - Write $q = \ell r$
- Suppose $\ell < q$ ($r \neq 1$)
  - Let f be a prime factor of r (and thus of q)
  - Then $g^{\frac{q}{f}} = g^{\frac{q}{r}\frac{r}{f}} = g^{\ell\frac{r}{f}} = 1^{\frac{r}{f}} = 1$
- Contrapositive:
  - $g^{\frac{q}{f}} \neq 1$ for each prime factor f of $q \Longrightarrow g$ has order q

This procedure requires knowledge of the factorization of q

Application: the "Oakley Groups" (RFC 2412 and 3526)

Standardized Groups for the Masses

$$p = 2^{2048} - 2^{1984} - 1 + 2^{64} \times \left([2^{1918}\pi] + 124476\right)$$
$$g = 2$$

Claim : g has order $p-1$ modulo p

Proof.

- Let q denote the order of g
- $\ell = (p-1)/2$ is also prime
  - p is a *Sophie Germain* prime or a *safe* prime
- Therefore $q \in \{2, \ell, 2\ell\}$
- $g^2 \neq 1$ and $g^\ell \neq 1$, therefore g has order $p-1$

$\square$

Conclusion: $\mathbb{Z}_p^\times = \langle 2 \rangle$

Creating Generators of Prime Order in $\mathbb{Z}_p^\times$ — Schnorr's Trick

Procedure

1. Choose a 256-bit prime q
2. Pick a random 1792-bit integer k
3. Set $p = 1 + kq$
4. If p is not prime, go back to 2.
5. Pick a random x modulo p
6. Set $g \leftarrow x^k$
7. If $g = 1$, go back to 5.
8. g has (prime) order q modulo p

Proof.

- $g^q = x^{p-1} = 1$
  - By Fermat's little theorem
- Therefore, if $g \neq 1$, then g has order q
  - cf. previous slides (easy case: q is prime) □

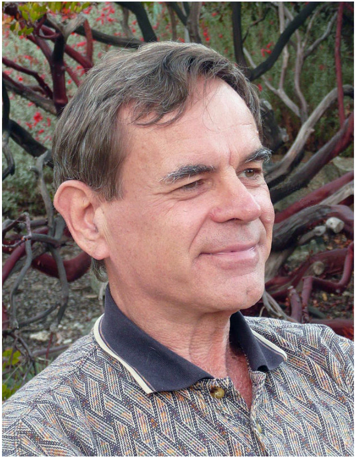
Digression: Primality Certificates
1975

If g has order $n-1$ modulo n, then n is prime

- $\langle g \rangle \subseteq \mathbb{Z}_n^\times$
- g has order $n-1$, *therefore* $|\mathbb{Z}_n^\times| = n-1$
- All integers except zero are invertible modulo n
- n does not have any non-trivial divisor
- n is prime

- providing g of order $n-1$ proves that n is prime
- Checking the order of g requires the factorization of $n-1$
- Certificate of n =
    1. g
    2. Factorization of $n-1$
    3. Certificates of the prime factors (recursively)
- Conclusion: PRIMES $\in$ NP

Digression: Primality Certificates
1975



Vaughan Pratt
(1944–)