CO 485/685 Fall 2022:
Lecture Notes

Lecture notes taken, unless otherwise specified, by myself during the Fall 2022 offering of CO 485/685, taught by David Jao.

Chapter/lecture titles are made-up nonsense and do not follow the textbook or any other published resource. Actually, scratch that, this entire document is nonsense because I am literally auditing this course two nested prerequisites behind.

Chapter 1

Introduction to Cryptography

Lecture 1 (09/07; skipped)

Lecture 2 Almost-Public Key Cryptosystems (09/09)

- For a symmetric key cryptosystem, require sets of key space K, message space M, and ciphertext space C
  - Define encryption function $Enc : K \to M \to C$ and decryption $Dec : K \to C \to M$
  - Correctness property: for all k, $Dec(k)$ is a left inverse of $Enc(k)$
  - Symmetric means that both decryption and encryption use shared secret k, which we assume is drawn randomly from K
- Public key encryption scheme (Diffie, Hellman, Merkle, c. 1976)
  - Setup similar: message space M and ciphertext space C but with two key spaces K_1 of public keys and K_2 of private keys
  - Define $Enc : K_1 \to M \to C$ and $Dec : K_2 \to C \to M$
  - Define $KeyGen : \mathbb{1}^\ell \to R \subset K_1 \times K_2$
    * For some reason, let $\mathbb{1}^n$ be the unary representation of n??
  - Correctness: for all $(k_1, k_2) \in R$ related, $Dec(k_2)$ is a left inverse of $Enc(k_1)$
- Merkle puzzle (1974)
  - Each party creates "puzzle" which is hard to solve but not too hard
  - Alice generates 1,000,000 puzzles and sends them to Bob
  - Bob solves one of the puzzles arbitrarily and sends half of the answer to Alice
  - Alice knows the answer, so Alice knows the second half of the answer, which becomes the shared secret
  - Eve cannot (realistically) solve 500,000 puzzles in time to intercept
- Diffie–Hellman key exchange
  - Consider the multiplicative group $G = (\mathbb{Z}/p\mathbb{Z})^* = 1, \ldots, p-1$ and some arbitrary element $g \in G$ with sufficiently large order
  - Alice privately picks some $x \in \mathbb{Z}$, computes g^x, and sends it to Bob
  - Bob privately picks some $y \in \mathbb{Z}$, computes g^y, and sends it to Alice
  - Both can now calculate a shared secret $k = g^{xy} = (g^x)^y = (g^y)^x$
  - Eve would have to solve the Diffie–Hellman problem: given p, g, g^x, g^y, find g^{xy} which is known to be hard
- Clifford Cocks privately discovered RSA 1973, DH 1974 for GCHQ (if you believe the intelligence community)

Lecture 3 A Public Key Cryptosystem – RSA (09/12)

- RSA (Rivest, Shamir, Adleman 1977): first cryptosystem and remains secure
- Theoretically secure, but implementations are ass (cf. "Fuck RSA")
- MATH 135 review of the algorithm:
  - This "textbook RSA" has practical flaws and is insecure
  - $KeyGen : \mathbb{1}^\ell \to (pk, sk) \in R$
    1. Choose random primes $p, q \approx 2^\ell$ where p and q are odd and distinct
    2. Compute $n = pq$
    3. Choose $e \in (\mathbb{Z}/\phi(n)\mathbb{Z})^\times$ where $\phi(n) = (p-1)(q-1)$
    4. Compute $d = e^{-1} \bmod \phi(n)$
    5. Disclose public key (n, e) and keep secret key (n, d)
  - $Enc : K_1 \to M \to C : (n, e) \mapsto m \mapsto m^e \bmod n$ where $M = (\mathbb{Z}/n\mathbb{Z})^\times = x : \mathbb{Z}/n\mathbb{Z} : \gcd(x, n) = 1 = C$
    - Weird that M depends on n (part of the key). In practice, it doesn't matter because the only messages that divide n are the primes, which breaks RSA anyways
  - $Dec : K_2 \to C \to M : (n, d) \mapsto c \mapsto c^d \bmod m$
- Correctness: Must show that $(m^e \bmod n)^d \bmod n = m$ *Proof.* $(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$ (exponentiation under mod). Then, since $d = e^{-1} \bmod \phi(n)$, there exists k such that $de - 1 = k\phi(n)$, we have $m^{\phi(n)k+1} \equiv (m^{\phi(n)})^k m \equiv m \pmod{m}$. This holds by Euler's theorem ($\forall m \in (\mathbb{Z}/n\mathbb{Z})^\times, m^{\phi(n)} \equiv 1 \pmod{n}$) or Fermat's Little Theorem + Chinese Remainder Theorem (MATH 135)
- Security: Trivial that factoring $n = pq$ breaks RSA by computing $\phi(n)$
  - Conversely, if you know $\phi(n) = (p-1)(q-1)$ you can take $q\phi(n) = (n-1)(q-1)$ and solve for q
    - To avoid this, use the Carmichael exponent $\lambda(n) = \operatorname{lcm}(p-1, q-1)$ instead of $\phi(n)$ which works. Of course, this doesn't work in practice because it's not actually that much different
  - For any non-trivial case, knowing one pair (e, d) also allows factoring n
  - Must make an assumption about hardness to prove security:
    - Factoring assumption: factoring random integers is hard
    - RSA factoring assumption: factoring $n = pq$ is hard (see, e.g., elliptical curve algorithm which depends on size of smallest prime in the factorization)
      - Of course, quantum computing fucks all of this to hell (see troll PQRSA which uses many small primes to make terabyte-sized moduli)
    - RSA assumption: given n, e, $m^e \bmod n$, it is hard to find m
  - Can prove RSA assumption $\implies$ RSA works (cannot prove without assumption without better results from complexity theory)

Lecture 4 Security Definitions (09/14)

- Security definitions, e.g., OW-CPA, IND-CPA, IND-CCA (Boneh, Shoup)
- How secure is a cryptosystem? Specify:
  - Allowable interactions between adversaries and parties
    - Second part of abbreviation
  - Computational limits of adversary
    - Not usually specified, usually probabilistic polynomial time

  - Goal of the adversary to "break" the cryptosystem
    * First part of abbreviation
- OW-CPA: "one-way chosen-plaintext attack"
  - Adversary, given public key pk and encryption c of message m under pk, wants to determine m
  - Formally, given a random pk and c such that $c = Enc(pk, m)$ for some random m, it is infeasible for any probabilistic polynomial time algorithm $\mathcal{A}$ to determine m with non-negligible probability. That is, $\Pr[\mathcal{A}(pk, c) = m] = O(\frac{1}{\lambda^c})$ for all $c > 0$.
- Easier way to formalize ("Sequences of Games", Shoup 2004)
  - Two players: challenger $\mathcal{C}$ and adversary $\mathcal{A}$
  - Then, OW-CPA is
    1. $\mathcal{C}$ runs $KeyGen : \mathbb{1}^\lambda \xrightarrow{\$} (pk, sk)$
    2. $\mathcal{C}$ chooses $m \xleftarrow{\$} M$
    3. $\mathcal{C}$ computes $c \leftarrow Enc(pk, m)$
    4. $m' \xleftarrow{\$} \mathcal{A}(pk, c)$
    * with the win condition that $m' = m$, and we say that a cryptosystem is OW-CPA if a probabilistic polynomial time adversary $\mathcal{A}$ cannot win this game with non-negligible probability
  - IND-CPA (Goldmeier, Micoli 1984): indistinguishability
    1. $\mathcal{C}$ runs $(pk, sk) \xleftarrow{\$} KeyGen(\mathbb{1}^\lambda)$
    2. $(m_0, m_1) \xleftarrow{\$} \mathcal{A}(\mathbb{1}^\lambda, pk)$
    3. $\mathcal{C}$ picks $b \xleftarrow{\$} 0, 1$
    4. $\mathcal{C}$ computes $c \xleftarrow{\$} Enc(pk, m_b)$
    5. $b' \xleftarrow{\$} \mathcal{A}(\mathbb{1}^\lambda, pk, c)$
    * with the win condition $b = b'$, and a cryptosystem is IND-CPA if for all prob. poly. time $\mathcal{A}$, $\left|\frac{1}{2} - \Pr[\text{win}]\right| = O(\frac{1}{\lambda^\varepsilon})$ for all $\varepsilon > 0$
    * Encryption function must be random, otherwise $\mathcal{A}$ can re-encrypt

Lecture 5 Actual IND-CPA systems (09/16)

- IND-CPA is the standard security definition for symmetric security
  - Ciphertext contains no information about plaintext (except length)
- Design a slightly different equivalent IND-CPA game:
  1. $\mathcal{C}$ runs $(pk, sk) \xleftarrow{\$} KeyGen(\mathbb{1}^\lambda)$
  2. $(m_0, m_1) \xleftarrow{\$} \mathcal{A}(\mathbb{1}^\lambda, pk)$
  3. $\mathcal{C}$ picks $b \xleftarrow{\$} 0, 1$
  4. $\mathcal{C}$ computes $c_1 \xleftarrow{\$} Enc(pk, m_b)$ and $c_2 \xleftarrow{\$} Enc(pk, m_{b-1})$
  5. $b' \xleftarrow{\$} \mathcal{A}(\mathbb{1}^\lambda, pk, c_1, c_2)$
- Consider textbook RSA: $\mathcal{A}$ can choose $m_0 \neq m_1$ and compute $Enc(pk, m_0)$ and $Enc(pk, m_1)$ which allows it to win
  - In general, this applies to any scheme with deterministic encryption
- Goldwasser-Micali ("Probabilistic Encryption" 1982)
  1. Pick $n = pq$ (useful to have $p \equiv q \equiv 3 \pmod 4$)
  2. Pick $r \in (\mathbb{Z}/n\mathbb{Z})^\times$ such that $r \not\equiv x^2 \pmod p$ and $r \not\equiv x^2 \pmod q$
  3. Define $pk = (n, r)$ and $sk = (p, q)$
  4. Select a message bit b from $M = 0, 1$

  5. Encrypt $Enc(b) = r^b y^2$ for some $y \xleftarrow{\$} (\mathbb{Z}/n\mathbb{Z})^\times$
  - Then, decrypt by determining ciphertext's squareness mod n
    * This is easy with the factorization $n = pq$ by Euler's criterion (a is square mod prime p if and only if $a^{(p-1)/2} \equiv 1 \pmod{p}$)
    * Determining squareness without factorization of n is hard, apparently
  - Since plaintexts are one bit, OW $\iff$ IND and this is provable under the circular-y assumption that determining squareness is hard
  - Also one bit messages are literally useless so who cares
- Elgamal (1984) (sometimes IND-CPA)
  - Publickeycryptosystemified Diffie-Hellman
  1. Setup is the same as DH, take some element $g \in G$ of a group
  2. Define $pk = g^x$ and $sk = x$
  3. Encrypt $Enc(m) = (g^y, g^{xy} \cdot m)$ for $y \xleftarrow{\$} \mathbb{Z}$
  - Then, decrypt $Dec(c_1, c_2) = \frac{c^2}{c_1^x} = \frac{g^{xy} \cdot m}{(g^y)^x} = m$
  - In general, key sharing schemes can be cryptosystemified like this
  - In an IND-CPA game, given $(g^y, g^{xy} m_b)$
    * Divide out m_0 to get either g^{xy} (if $m_b = m_0$) or garbage
    * Real challenge is distinguishing g^{xy} from garbage
  - Decisional Diffie-Hellman assumption: in the following game, $\left|\Pr[\mathcal{A} \text{ wins}] - \frac{1}{2}\right|$ is negligible in λ
    1. $\mathcal{C}$ chooses $p \xleftarrow{\$} \mathbb{Z}$ prime, $p \approx 2^\lambda$
    2. $\mathcal{C}$ chooses $g \in (\mathbb{Z}/p\mathbb{Z})^\times$
    3. $\mathcal{C}$ chooses $x, y \xleftarrow{\$} \mathbb{Z}$ and $h \xleftarrow{\$} (\mathbb{Z}/p\mathbb{Z})^\times$, computes $g_1 = g^x$, $g_2 = g^y$, $g_3 = g^{xy}$
    4. $\mathcal{C}$ chooses $b \xleftarrow{\$} 0, 1$ and $g_4 = g_3$ if $b = 0$ and h if $b = 1$
    5. $b' \leftarrow \mathcal{A}(\mathbb{1}^\lambda, p, g, g_1, g_2, g_4)$
  - Can prove: if DDH assumption holds, Elgamal is IND-CPA
- Layers of assumptions here:
  - DLOG: given g and g^x, it is hard to find x
  - CDH: given g, g^x, and g^y, it is hard to find g^{xy} (equivalent to Elgamal being OW-CPA)
  - DDH: given g^{xy} and garbage, is hard to distinguish the garbage
- How to piss off mathematicians: solving DLOG in $\mathbb{Z}/n\mathbb{Z}$ is easy but in $(\mathbb{Z}/p\mathbb{Z})^\times$ is hard
  - But $(\mathbb{Z}/p\mathbb{Z})^\times$ is isomorphic to $\mathbb{Z}/(p-1)\mathbb{Z}$ so DLOG difficulty must not be preserved over isomorphism
  - Specifically, DLOG is as exactly hard as computing the isomorphism (notice that we send $x \mapsto g^x$)
- DDH is actually easy in $(\mathbb{Z}/p\mathbb{Z})^\times$, need a subgroup $G \subset (\mathbb{Z}/p\mathbb{Z})^\times$ with $|G|$ prime

Chapter 2

Quadratic Residues

Lecture 6 Number Theory Background (09/19)

- Recall: RSA primes are gigantic so it takes time to do operations
  - e.g. picking $e \in (\mathbb{Z}/\phi(n)\mathbb{Z})^\times$ or finding $d = e^{-1} \pmod{\phi(n)}$ using EEA which runs in a logarithmic number of steps
  - e.g. running $Enc(m) = m^e \pmod{n}$ or $Dec(c) = c^d \pmod{n}$ using square-and-multiply which runs in a logarithmic number of steps
- Hard: picking non-squares in integers modulo p
  - Set of primes $\left|((\mathbb{Z}/p\mathbb{Z})^\times)^2\right| = \frac{p-1}{2}$ for odd $p > 2$
  - This is because $f(x) = x^2$ is a 2-to-1 function on $(\mathbb{Z}/p\mathbb{Z})^\times$
    - To prove, show $f(a) = f(b) \iff a = \pm b$
    - Apply Euclid's Lemma: $p \mid (x-y)(x+y)$ implies $p \mid x - y$ or $p \mid x + y$, equivalently, $x = y \pmod{p}$ or $x = -y \pmod{p}$
    - Also another theorem: for R integral domain, every polynomial of degree n over R has at most n roots

Lecture 7 Squares Under a Modulus (09/21)

The big problem: Given $(\mathbb{Z}/n\mathbb{Z})^\times$ and $x \in (\mathbb{Z}/n\mathbb{Z})^\times$, when is $x \equiv \square \pmod{n}$?

For example, for $\mathbb{Z}/15\mathbb{Z}$, 1 and 4 are squares; for 8: just 1; for 7: 1, 2, and 4; and for 13: 1, 3, 4, 9, 10, and 12.

This breaks down into cases: n composite, n prime power, n prime

Theorem

Suppose $n = \prod p_i^{e_i}$. Then, $x \equiv \square \pmod{n}$ if and only if for all i, $x \equiv \square \pmod{p_i^{e_i}}$.

Proof. Suppose $x = y^2 \pmod{n}$ for a unit y. Then, $n \mid (x - y^2)$ and $p_i^{e_i} \mid (x - y^2)$ by transitivity. That is, $x \equiv y^2 \pmod{p_i^{e_i}}$. In the reverse direction, if $p_i^{e_i} \mid (x - y^2)$ for all i, then by UPF (with some omitted detail), $n \mid (x - y^2)$. $\square$

The prime power case reduces to the prime case under condtions discovered in the homework problems lol.

Theorem

The number of squares in $(\mathbb{Z}/p\mathbb{Z})^\times$ is $\frac{p-1}{2}$ for primes $p \geq 3$.

Proof. This is because $x = y^2 = (-y)^2$ and the size of the set is $p-1$.

Build a table (x, g^x) instead of (x, x^2):

For $p = 13$ and $g = 2$, we get $(1, 2, 4, 8, 3, 6, 12 = -1, -2, -4, -8, -3, -6, -12 = 1)$ and the squares are the even-indexed values $(1, 4, 3, 12, 9, 10, 1)$.

This works for tables starting with non-squares: in fact, if $g \neq \square$, then $g^3 \neq \square$ (by the contrapositive, if $g^3 = \square$, then $g = \frac{g^3}{g^2} = \frac{\square}{\square} = \square$).

This gives us the result that $g^x = g^y$ when $x \equiv y \pmod{p-1}$ (note that this is equivalent to Fermat's Little Theorem, the reverse direction requires g coprime to $p-1$). $\square$

Definition (*order*)

$\operatorname{ord}(a)$ is the period of $x \mapsto a^x$ for $a \in (\mathbb{Z}/p\mathbb{Z})^\times$.
Equivalently, $\operatorname{ord}(a) = \min\{t \in \mathbb{Z} : a^t = 1, t > 0\}$.

Lemma

Given elements a and b, numbers x and y:
- $a^x = 1$ if and only if $\operatorname{ord}(a) \mid x$
- $a^x = a^y$ if and only if $x \equiv y \pmod{\operatorname{ord}(a)}$
- $\operatorname{ord}(a^x) = \frac{\operatorname{ord}(a)}{\gcd(x, \operatorname{ord}(a))}$
- If $\operatorname{ord}(a)$ and $\operatorname{ord}(b)$ are coprime, then $\operatorname{ord}(ab) = \operatorname{ord}(a)\operatorname{ord}(b)$.

Proof. Only prove the last one:

Let $t = \operatorname{ord}(a)$, $u = \operatorname{ord}(b)$, $v = \operatorname{ord}(ab)$. Then, $(ab)^{tu} = a^{tu}b^{tu} = 1^u 1^t = 1$ so we have $v \mid tu$. Now, WLOG, $(ab)^{vu} = 1^u = 1 \implies a^{vu}b^{vu} = a^{vu}1 = a^{vu} = 1$. This gives $t \mid vu$ and $t \mid v$ since $\gcd(t, u) = 1$. Likewise, $u \mid v$ and we can conclude $tu \mid v$ because $\gcd(t, u) = 1$. That is, $tu = v$. $\square$

Lecture 8 Squares cont'd (09/23)

Definition (*primitive element*)

$g \in G$ where $\{g^n : n \in \mathbb{N}\} = G$. Also called a generator.

Recall: if there exists primitive $g \in (\mathbb{Z}/p\mathbb{Z})^\times$, then for all $h \in (\mathbb{Z}/p\mathbb{Z})^\times$ where $h = g^k$, $h \equiv \square \iff k$ even. We can determine squareness using this fact, but finding k such that $h = g^k$ is doing a discrete log, which is hard.

Whether or not a primitive element exists is a non-trivial observation:

Theorem (*Gauss' primitive root*)

For all primes p, $(\mathbb{Z}/p\mathbb{Z})^\times$ has a primitive element.

Proof. Observe that for all polynomials $f(x) \neq 0$ over $\mathbb{Z}/p\mathbb{Z}$, the number of roots of $f(x)$ is at most $\deg f$. Note that factorization fails in $\mathbb{Z}/n\mathbb{Z}$ in general: e.g. $x^2 - 1 = (x-1)(x+1) = (x-3)(x-5) \bmod 8$ or something weird like $x = (3x+2)(2x+3) \bmod 6$. We have this observation because $\mathbb{Z}/p\mathbb{Z}$ is an integral domain (and indeed, a field).

Consider $a \in (\mathbb{Z}/p\mathbb{Z})^\times$.

Claim $t = \operatorname{ord}(a) \mid p-1$. Write $p-1 = tq + r$. If $r = 0$, done. If $r > 0$, $\operatorname{ord}(a) = r < t$, contradiction and indeed $r = 0$.

For each divisor d of $p-1$, consider $S_d = \{x \in (\mathbb{Z}/p\mathbb{Z})^\times : \operatorname{ord}(x) = d\}$. Then, $\bigcup_{d|p-1} S_d = (\mathbb{Z}/p\mathbb{Z})^\times$ and this is a disjoint union. To prove Gauss' theorem, we just need $\left|S_{p-1}\right| > 0$.

Proceed in general for arbitrary $|S_d| > 0$ for all $d \mid p-1$.

If $S_d = \varnothing$, then $|S_d| = 0$. Otherwise, claim that $|S_d| = \phi(d) = |(\mathbb{Z}/d\mathbb{Z})^\times|$.

If S_d is not empty, then $\exists a \in S_d$ where $\operatorname{ord}(a) = d$. Consider $x^d - 1$. The roots of this polynomial will include all elements of S_d (and others). We can write the set of roots as exactly $\{a^0, \dots, a^{d-1}\}$. So for all $b \in S_d$, $b = a^k$ since b is a root and we need only count those powers with order d. But that is exactly $\operatorname{ord}(a^i) = \frac{\operatorname{ord}(a)}{\gcd(i,d)} = \frac{d}{\gcd(i,d)}$. So we are counting the i such that $\gcd(i, d) = 1$, which is exactly $\phi(d)$.

Now, $p - 1 = |(\mathbb{Z}/p\mathbb{Z})^\times| = \left|\bigcup_{d|p-1} S_d\right| = \sum |S_d| \leq \sum \phi(d)$ which is equal to $p-1$ by Möbius inversion. That last inequality being an equality implies that $|S_d| \neq 0$ for any $d \mid p-1$, and in particular $p-1 \mid p-1$.

Quick combinatorical proof of this fact: write out all the $p-1$ fractions over $p-1$, then each of $\phi(d)$ is the number of fractions where the denominator reduces to d. The sum must be $p-1$. $\square$

Lecture 9 Applying to DDH (09/26)

Recall the Decisional Diffie-Hellman problem: Given g, g^x, g^y, g^z, determine if $z = xy$. Formally, as a game:

- $\mathcal{C}$ chooses a bit $b \in \{0, 1\}$ and $x, y \xleftarrow{\$} \mathbb{Z}$
- $b' \leftarrow \mathcal{A}(g, g^x, g^y, g^z)$ where $z \leftarrow \begin{cases} xy & b = 0 \\ \xleftarrow{\$} \mathbb{Z} & b = 1 \end{cases}$
- Win condition: $b = b'$ with non-negligible probability

Notice that if g is a primitive root, then $|\{g^x : x \in \mathbb{Z}\}| = p-1$. But bruteforce DLOG takes $\frac{p-1}{2}$ steps on average. Then, Elgamal is IND-CPA $\iff$ DDH holds.

Proposition

> The Decisional Diffie-Hellman assumption in $(\mathbb{Z}/p\mathbb{Z})^\times$ with a primitive base g does not hold.

Proof. We tell squares and non-squares apart.

Recall from last lecture's theorem we have that if g is a primitive root, $g^x \equiv \square \pmod p \iff x \equiv 0 \pmod 2$. Then, by Euler's criterion, $a \equiv \square \pmod p \iff a^{(p-1)/2} \equiv 1 \pmod p$. Therefore, it is possible to tell the parity of x, y, and z in reasonable time using Euler's criterion (since raising to a power is easy).

If xy is odd only when x and y are odd, so if you know the parity of z you can distinguish if $z = xy$ or random with non-negligible advantage. $\square$

Proposition (*Euler's criterion*)

$a \equiv \square \pmod p \iff a^{(p-1)/2} \equiv 1 \pmod p$

Proof. Suppose $a \equiv \square$ iff $a \equiv g^k$ for even $k = 2\ell$ iff $a^{(p-1)/2} = (g^k)^{(p-1)/2} = g^{k(p-1)/2} = (g^{p-1})^\ell = 1^\ell = 1$ by FℓT.

Otherwise, $a \not\equiv \square$ iff $a = g^k$ for $k = 2\ell + 1$ iff $a^{(p-1)/2} = (g^k)^{(p-1)/2} = g^{(p-1)/2 \cdot (2\ell+1)} = g^{(p-1)/2 \cdot 2\ell} \cdot g^{(p-1)/2} = g^{(p-1)/2} \neq 1$. But in fact $g^{(p-1)/2} = \sqrt{g^{p-1}} = \sqrt{1} = -1$ since it is not positive 1. $\square$

Corollary. For $p > 2$, -1 is a square mod p if and only if $p \equiv 1 \pmod 4$.

Proof. For -1 to be a square, we need $(-1)^{(p-1)/2} \equiv 1 \pmod p$. That is, $\frac{p-1}{2}$ is even and we have $p \equiv 1 \pmod 4$. $\square$

This quantity $g^{(p-1)/2}$ is useful and we give it a name:

Definition (*Legendre symbol*)

For $p > 2$ and $a \in \mathbb{Z}/p\mathbb{Z}$, the quadratic character of a, written $(\frac{a}{p}) = a^{(p-1)/2}$, is 1 if $a \equiv \square$, 0 if $a \equiv 0$, and -1 if $a \not\equiv \square$.
Equivalently, define $\chi_p : (\mathbb{Z}/p\mathbb{Z})^\times \to \{\pm 1\} : a \mapsto (\frac{a}{p})$ and notice that this is a multiplicative homomorphism that preserves $\chi_p(ab) = \chi_p(a)\chi_p(b)$.

Theorem (*multiplicativity*)

$(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$

Proof. $(\frac{ab}{p}) = (ab)^{(p-1)/2} = a^{(p-1)/2} b^{(p-1)/2} = (\frac{a}{p})(\frac{b}{p})$ $\square$