12 Additional Material

12.1 Coin Flipping

---

1. **Introduction:** The whole idea of "coin flipping" is for two parties to agree that a "coinflip" is fair when they are not in the presence of the coin.

2. **Theorem:** Suppose p, q are distinct odd primes. Let $A \neq 0 \bmod n = pq$. If $x^2 \equiv A \bmod n$ has any solutions (if at all) then it has 4 distinct solutions mod n.

   Proof. First note that if $x^2 \equiv A \bmod n$ then $x^2 \equiv A \bmod p$ and $x^2 \equiv A \bmod q$. Then we know that $x^2 \equiv A \bmod p, q$ have exactly two solutions each. Next observe that by the CRT, solutions to $x^2 \equiv A \bmod n$ correspond exactly to pairs of solutions to $x^2 \equiv A \bmod p, q$. Why are they distinct?

   - Suppose $x \equiv a \bmod n$ is one solution. So $a^2 \equiv A \bmod n$. Then, consider the system

     $$x \equiv a \bmod p$$
     $$x \equiv a \bmod q$$

     by the CRT this has a unique solution mod $pq = n$. This solution will satisfy $x^2 \equiv a^2 \equiv A \bmod p$ and $x^2 \equiv a^2 \equiv A \bmod q$. Since $\gcd(p, q) = 1$ we have that $x^2 \equiv A \bmod pq = n$. Call this solution X.

     Note that $x = -X$ is a solution as well since $(-X)^2 \equiv X^2 \equiv A \bmod n$. Moreover, note that $x = -x$ satisfies the system

     $$x \equiv -a \bmod p$$
     $$x \equiv -a \bmod q$$

   - Now, consider this system,

     $$x \equiv a \bmod p$$
     $$x \equiv -a \bmod q$$

     by the CRT this has a unique solution mod $pq = n$. Call this solution Y.

     likewise, $x = -Y$ is a solution as well. Since it satisfies

     $$x \equiv -a \bmod p$$
     $$x \equiv a \bmod q$$

So, all together we have $x = X, -X, Y, -Y$ as our solution where they are all distinct mod $n = pq$. □

3. **Theorem:** If $p \equiv 3 \bmod 4$ and if $x^2 \equiv A \bmod p$ has solutions, we can find them easily.

   Proof. We know if it has any, it has two. Since A is a QR we know that $\left(\frac{A}{p}\right) = 1$ and then,

   $$\left(\pm A^{\frac{p+1}{4}}\right)^2 \equiv A^{\frac{p+1}{2}} \equiv A \cdot A^{\frac{p-1}{2}} \equiv A\left(\frac{A}{p}\right) \equiv A \cdot 1 \equiv A \bmod p$$

   So we know that $x = \pm A^{\frac{p+1}{4}}$ to be the two solutions. □

4. **Theorem:** Consider $x^2 \equiv A \bmod n$, if $n = pq$ and we know p, q and if there are solutions, there are 4 and we can find them easily.

   Proof. Since $x^2 \equiv A \bmod n$ has solutions so do $x^2 \equiv A \bmod p$ and $x^2 \equiv A \bmod q$. We can find these as we have seen. They are $x \equiv \pm A^{\frac{p+1}{4}} \bmod p$ and $x \equiv \pm A^{\frac{q+1}{4}} \bmod q$. This leads us to the 4 systems in the CRT.

   - We solve,

     $$x \equiv A^{\frac{p+1}{4}} \bmod p$$
     $$x \equiv A^{\frac{q+1}{4}} \bmod q$$

     Call that result X, this also gives us $-X$.
   - We solve,

     $$x \equiv A^{\frac{p+1}{4}} \bmod p$$
     $$x \equiv -A^{\frac{q+1}{4}} \bmod q$$

     Call that result Y, this also gives us $-Y$.

   So we have 4 solutions. □

   Ex. Supose $p = 31, q = 43$ so $n = pq = 1333$. Suppose we know $x^2 \equiv 669 \bmod 1333$ has solutions. Find them!

   - Solve,

     $$x \equiv 669^{(31+1)/4} \equiv 7 \bmod 31$$
     $$x \equiv 669^{(43+1)/4} \equiv 14 \bmod 43$$

     Which gives us $X = 100 \bmod 1333$ and $-X = 100 \bmod 1333$.

- Solve

$$x \equiv 669^{(31+1)/4} \equiv 7 \bmod 31$$
$$x \equiv -669^{(43+1)/4} \equiv -14 \bmod 43$$

Which gives us $Y = 1061 \bmod 1333$ and $-X = -1061 \equiv 272 \bmod 1333$.

So our 4 solutions are; $100, -100, 1061, 272$.

5. **Theorem:** Knowing one of $\pm X$ and one of $\pm Y$ is equivalent to factoring n.

Proof.
$\rightarrow$ Suppose we know X and Y. Observe that $X + Y \equiv a + a \equiv 2a \bmod p$ (we know that $2a \not\equiv 0 \bmod p$ since $p \nmid 2$ and $p \nmid a$) and $X+Y \equiv a+(-a) \equiv 0 \bmod q$. So $q \mid (X + Y)$ and $p \nmid (X + Y)$, since if $p \mid (X + Y$ then $p \mid 2a$ but then $p \mid a$ so $a \equiv 0 \bmod p$. Which leads to a contradiction. So $\gcd(X + Y, n) = q$, thus we can find q and then p follows as $p = \frac{n}{q}$. Similar arguments work for knowing X and $-Y$, $-X$ and Y, $-X$ and $-Y$.

$\leftarrow$ This is obvious, we did it above. $\square$

6. **Process:**
   (a) Alice picks primes p, q both congruent to 3 mod 4, both of which are distinct and odd. She finds $n = pq$. She sends n to Bob.
   (b) Bob picks $0 < b < n$ and calculates $S \equiv b^2 \bmod n$. He knows that $X^2 \equiv S$ has 4 solutions but he can't find all of them since he can't factor n. He only has two solutions, which are b and $-b$. Bob then sends S back to Alice.
   (c) Alice finds the 4 solutions to $x^2 \equiv S \bmod n$. She gets $X, -X, Y, -Y$, one of these corresponds to Bob's b but she does not know which.
   (d) Alice chooses one and sends it back to Bob.
   (e) If she sends back $\pm b$, it does not help Bob. However, if she sends back either of the others, he can factor n. If Bob can factor n he wins! (50% chance that he gets an integer that helps him factor n.)

7. **Ex.** Alice chooses $p = 31$ and $q = 43$ so $n = 1333$. She sends 1333 to Bob. Bob chooses $b = 100$, and finds $S \equiv b^2 \equiv 669 \bmod 1333$. He then sends it to Alice, Alice solves $x^2 \equiv 669 \equiv 669 \bmod 1333$. She gets $100, 272, 1061, 1233$ as the solutions. Alice knows that Bob's b corresponds to one of these, but she has no way of determining which one that is. She then picks one and sends it to Bob. If she sends back $100, 1233 (\equiv -100)$ Bob can't factor n. If she sends back $272, 1061$ Bob can factor n, since $\gcd(100 \pm 272, 1333)$ and $\gcd(100 \pm 1061, 1333)$ will give him either 31 or 43.

12.2 El-Gamal Cryptosystem

1. **Introduction:** This system is based on the difficulty of calculating discrete logarithms. Like RSA this is asymmetric.

2. **Key Creation:** Bob chooses one large prime p, a primitive root r mod p, and an integer a with $1 \leq a \leq p-2$. He keeps a secret. He then calculates $b \equiv r^a \bmod p$. Then he makes (p, r, b) public. Observe that $a \equiv \text{ind}_r b \bmod p-1$ (this is extremely difficult to calculate).

3. **Encryption:** Suppose Alice wishes to send the plaintext block P to Bob. She first chooses a random integer k with $1 \leq k \leq p-2$, then she encrypts via $\epsilon(P) \equiv (r^k, Pb^k) \bmod p$. This produces a pair (γ, δ) which is the ciphertext, i.e. $\gamma \equiv r^k \bmod p$ and $\delta \equiv Pb^k \bmod p$.

   Note: By choosing a different (randomly) k each time, we can ensure that the same P, if encrypted multiple times, will yield different ciphertext. Which can alleviate issues of frequency analysis.

4. **Decryption:** Bob recieves (γ, δ), we claim that $P \equiv \gamma^{p-1-a}\delta \bmod p$ (note that $p-1-a \geq 1$). To see this note:

$$\begin{aligned}\gamma^{p-1-a}\delta &\equiv (r^k)^{p-1-a}(Pb^k) \bmod p \\ &\equiv (r^{p-1})^k (r^a)^{-k} b^k P \bmod p \\ &\equiv (1^k)(b^{-k})b^k P \bmod p\end{aligned}$$

   mult. inverse of $r^a \exists$ since p is prime and... Some notes about the derivation above, we know $(r^a)^{-k}$ since it is the multiplicative inverse of r^a which we knows exists since p is prime, furthermore we know that $r^a \equiv b$ and $b \leq p-1$ and p is prime. For $(r^{p-1})^k$ we have that $r^{p-1} \equiv 1$ since r is a primitive root. Thus we have $\epsilon^{-1}(\gamma, \delta) \equiv \gamma^{p-1-a}\delta \bmod p$.

5. **Ex.** Bob selects $p = 2539$ and $r = 2$ (a PR) and $a = 42$ (kept private). Bob calculates $b = 2^{42} \equiv 1305 \bmod 2539$, so his public key is $(p, r, b) = (2539, 2, 1305)$. Eve knows $2^a \equiv 1305 \bmod 2539$ but she can't find a (this is the problem that is hard to solve). Alice wants to send `OHNO`, she breaks it into 2 blocks of size 2; `OH`=1407, `NO`=1314. Then she encrypts:

   1407: She chooses $k = 100$ then she does

$$\begin{aligned}\epsilon(1407) &= (2^{100}, 1407 \cdot 1305^{100}) \bmod 2359 \\ &= (613, 635) \bmod 2539\end{aligned}$$

   1314: She chooses $k = 200$ then she does

$$\begin{aligned}\epsilon(1314) &= (2^{200}, 1314 \cdot 1305^{200}) \bmod 2359 \\ &= (2356, 1494) \bmod 2539\end{aligned}$$

So she then sends (613, 635) and (2356, 1494).

Bob then gets those and he decrypts:

(613, 635): $\epsilon^{-1}(613, 635) \equiv 613^{2539-1-42} \cdot 635 \equiv 1407 \bmod 2539$.
(2356, 1494): $\epsilon^{-1}(2356, 1394) \equiv 2356^{2359-1-42} \cdot 1494 \equiv 1314 \bmod 2539$.
Then he is done.

6. **Comments:**
   (a) To decrypt we need a, and getting this from b is hard.
   (b) In theory we could try a^x for lots of x but,
   (c) Use different k each time so if $P_1 = P_2$ we get $c_1 \neq c_2$.
   (d) Ciphertext is two times longer than the plaintext, which is a disadvantage, but it has security given above.
   (e) Typically ElGamel and RSA (asymmetric encrpytion schemes) are actually not used for the entire message. They are used to exchange a symmetric key which is then used to transmit data since it is fast.
   (f) Signing messages is possible but not as easy. Alice can not simply use her own d since there is no obvious mechanism for getting a random k involved.
   (g) While verifying primitive roots can be hard, Bob's PR need not be public. Alice could give it to him, in fact so could Eve!
   (h) Alice should definetly use a different (random) k each time. If Alice uses the same k for P_1 and P_2 then *if* Eve figures out P_1 she can figure out P_2. This is because Eve will know $\gamma_1 \equiv P_1 b^k$ and $\gamma_2 \equiv P_2 b^k$ then observe that $P_2 \equiv \gamma_2 (b^k)^{-1} \equiv \gamma_2 \left(\gamma_1 P_1^{-1}\right)^{-1} \equiv \gamma_2 \gamma_1^{-1} P_1 \bmod p$.

12.3 Homomorphic Encryption

---

1. **Idea in Abstract:** Suppose Bob has data and needs calculations done with it. He wants Alice to do the calculation for him, but at the same time he doesn't want Alice to understand what she is doing. So, what Bob wants to do is to encrypt data, send it to Alice, she does the calculation on the encrypted data without decrypting it, and return the result to Bob so he can decrypt the answer.

2. **Basic Level/Goal:** Can we at least do it with addition and multipliction?

3. **Simple Obfuscation of Data:** Suppose Bob wants Alice to calculate AB with $A, B \in \mathbb{Z}^+$ without knowing A, B. He defines an encryption function $\epsilon(x) = \lg x$. He calculates $\epsilon(A) = \lg A$ and $\epsilon(B) = \lg B$, which he sends to Alice with the simple instruction: `Add these`. She does $\lg A + \lg B$ and returns

it. Bob uses $D(x) = 2^x$, since $D(\lg A + \lg B) = 2^{\lg A + \lg B} = 2^{\lg A} 2^{\lg B} = AB$. Provided Alice does not know what is going on, Bob is safe.

Note: This allows only AB, not $A + B$. If he wanted only AB, he could do $\epsilon(x) = 2^x$ and $D(x) = \lg x$. Then he tells Alice to multiply. Can he find an ϵ and D which allows both? This is symmetric—knowing $\epsilon \equiv$ knowing D.

4. **Rings & Ring Homomorphisms and Isomorphisms:**

   (a) **Definition:** A ring (loosely speaking) is a set of numbers (objects) with two operations, typically addition and multiplication.
   Ex.
   - $\mathbb{Z}_6 = \{0, 1, 2, 3, 4, 5\}$ with addition and multiplication mod 6.
   - $2\mathbb{Z}$ = even integers.
   - $\mathbb{C}$
   - $M_2\mathbb{R}$ = set of 2×2 matrices with entries in $\mathbb{R}$.
   - $\mathbb{Z}_2 \times \mathbb{Z}_3 = \{(x, y) \mid x \in \mathbb{Z}_2, y \in \mathbb{Z}_3\}$ operations are componentwise. The first is mod 2, and the second mod 3. So $(1, 2) + (0, 2) = (1 + 0, 2 + 2) = (1, 1)$.

   (b) **Definition:** Given two rings R and S a ring homomorphism is a mapping (function) $\phi : R \to S$ with two properties;

   $$\phi(xy) = \phi(x)\phi(y)$$

   $$\phi(x + y) = \phi(x) + \phi(y)$$

   Ex. Define $\phi : \mathbb{Z} \to \mathbb{Z}_6$ by $\phi(x) = x \bmod 6$. So $\phi(3) = 3$, $\phi(10) = 4$, etc... Observe then that

   $$\phi(xy) = xy \bmod 6 = (x \bmod 6)(y \bmod 6) = \phi(x)\phi(y)$$

   $$\phi(x + y) = (x + y) \bmod 6 = (x \bmod 6) + (y \bmod 6) = \phi(x) + \phi(y)$$

   (c) **Definition:** A ring isomorphism is a ring homomorphism which is one-to-one and onto.
   Ex. Our example above is not since it is not one-to-one, $\phi(0) = \phi(6) = \phi(12) = \cdots$.
   Ex. $\phi : \mathbb{Z}_6 \to \mathbb{Z}_2 \times \mathbb{Z}_3$ by:

   $$\phi(0) = (0, 0), \quad \phi(1) = (1, 1), \quad \phi(2) = (0, 2)$$
   $$\phi(3) = (1, 0), \quad \phi(4) = (0, 1), \quad \phi(5) = (1, 2)$$

   This mapping is a ring isomorphism (this is not obvious). We can see that it's one-to-one and onto. The two properties are not obvious but lets verify an example with 3 and 5.

   $$\phi(3 + 5) = \phi(2) = (0, 2) = (1, 0) + (1, 2) = \phi(3) + \phi(5)$$

$$\phi(3 \cdot 5) = \phi(3) = (1,0) = (1,0)(1,2) = \phi(3)\phi(5)$$

Bob could use this ϕ as an encryption function, he would set $\epsilon = \phi$ and $D = \phi^{-1}$. Bob wants Alice to do $3 \cdot 5 + 2$. He calculates $\{\epsilon(3), \epsilon(5), \epsilon(2)\} = \{(1,0), (1,2), (0,2)\}$ and sends to Alice with instructions to multiply the first two and add the third. Alice then does $(1,0)(1,2) + (0,2) = (1,0) + (0,2) = (1,2)$ and sends it back to Bob. Bob then does $D(1,2) = 5$ and he gets his result, $3 \cdot 5 + 2 = 5 \in \mathbb{Z}_6$.

5. **Noise:** In some encryption methods (Lattic Based methods) some noise is added to the encrypted data as part of the process. That noise can be corrected for during decryption. We've actually seen this (sort of) as a side effect, we just never actually did it.

   Recall our $\epsilon(x) = \lg x$. Bob wants Alice to calulcate $3 \cdot 5$, we said he would send her $\lg 3$ and $\lg 5$ ... what does he actually send her? "$\lg 3$"? That gives away our encryption function. What about $\lg 3 = 1.584962500721156\ldots$, how many digits does he send her?

   Suppose he sends 1.58 for $\lg 3$ and 2.32 for $\lg 5$. Alice calculates $1.58 + 2.32 = 3.90$ and sends it to Bob. Bob then takes 3.9 and raises it base 2, $2^{3.9} = 14.92852786\ldots \approx 15$. Can he conclude that two digits beyond the decimal point is good enough? Let's see

   Define $R(x, n)$ to be x rounded to the closest decimal with n digits beyond the decimal, i.e. $R(1.584, 2) = 1.58$. With this notation we have Bob's encryption/decryption functions as

   $$\epsilon(x) = R(\lg x, 2)$$

   $$D(x) = R(2^x, 0)$$

   Lets look at $3^2 \cdot 5 \cdot 7$, Bob calculates his two digit approximations

   $$\epsilon(3) = R(\lg 3, 2) = 1.58$$
   $$\epsilon(5) = R(\lg 5, 2) = 2.32$$
   $$\epsilon(7) = R(\lg 7, 2) = 2.81$$

   He then sends (1.58, 1.58, 2.32, 2.81) to Alice with instructions to sum them all. She does so and gets 8.29 which she sends back to Bob who then uses his decryption function.

   $$D(2^{8.29}) = R(2^{8.29}, 0) = 313$$

   This is wrong. The correct answer is $3^2 \cdot 5 \cdot 7 = 315$.

   What we're seeing here is as the calculations include more and more operations (complicated) the more complex the noise accumulates. A few ways to adjust for this are:

(a) Bob could only send simple calculations.

(b) Bob could pre-emptively determine how bad the noise could be and adjust what he sends accordingly.

(c) Bob could give Alice some obscure instructions which would allow her to correct for the noise without realizing that she is correcting for the noise.

6. **Relation to Other Cryptosystems:**

   (a) RSA is partially homomorphic, meaning it works for one operation. We have $\epsilon(x) = x^e \bmod n$ Observe then that $\epsilon(xy) \equiv (xy)^e \bmod \text{n} \equiv (x^e \bmod n)(y^e \bmod n) \equiv \epsilon(x)\epsilon(y)$. But $\epsilon(x+y) \neq \epsilon(x) + \epsilon(y) \bmod n$.

   (b) El Gamal is partially homomorphic. We have $\epsilon(x) = (r^k, xb^k) \bmod p$ provided we understand that xy is encrypted using $k_x + k_y$ where kx is the random for x and ky is the random for y. Then,

$$\epsilon(xy) = (r^{k_x+k_y}, (xy)b^{k_x+k_y}) \equiv (r^{k_x}r^{k_y}, (xy)b^{k_x}b^{x_y}) \equiv \epsilon(x)\epsilon(y) \bmod P$$

   (c) Shift ciphers and affine ciphers and not even partially homomorphic. These systems work with neither addition or multiplication.

   (d) The Benaloh Cryptosystem works by doing $\epsilon(x) = g^x u^k$. Here we get:

$$\epsilon(x+y) = g^{x+y}u^{k_x+k_y} \equiv g^x g^y u^{k_x} u^{k_y} \equiv \epsilon(x)\epsilon(y) \bmod n$$

   Which maps addition to multiplication, making Benaloh a partially homomorphic system.

7. **Lattice-Based Cryptography:** This was first developed in the mid 2000s (super recent). Lattice-Based is nice because factoring and discrete logarithms can be done in polynomial time (quickly) on a quantum computer, but Lattice problems... it is thought that the problems are quantum resistant.

   Lattice-Based Problem: Consider the basis of $\mathbb{R}^2$ given by $[86, 21]$ and $[79, -85]$. What is the shortest vector you can build using integer combinations of these? Now try this in $\mathbb{R}^{10000}$..., this is the Shortest Vector Problem.

8. **Brief History:**

   (a) Up until the early 2000s it was not clear that asymmetric homomorphic encryption could even exist.

   (b) In about 2005 Lattice-based cryptography appears and people go "hmm". Thinking that this could be used to build a homomorphic encryption scheme.

   (c) In 2009 Craig Gentry (PhD Thesis) confirms that fully homomorphic encryption schemes can be built upon Lattices. But this has a lot of noise with it.

   (d) The current state of research is trying to reduce noise as well as tweaking these homomorphic schemes.

12.4 Problems

1. Use Pollard's Rho method to obtain a factor of each of the following. Use $x_0 = 2$ and $x_{n+1} = x_n^2 + 1$.

   (a) $n = 143$

   (b) $n = 5473$

   (c) $n = 234643$

2. From the context of class and notes, show that knowing X and $-Y$ is enough to factor n.

3. Emulate/rewrite the final coin-flip example from class with $p = 67$, $q = 83$, and $b = 123$. Describe Alice's choices, Bob's choice, who sends what to whom, the equation Alice solves, what those solutions are, and what possibilities might emerge.

   For the equation Alice solves write down the details as in the example of Theorem 3 in the notes but you can use technology to do the gritty calculations.

4. These relate to the El-Gamal Cryptosystem

   (a) Choose a prime p with $2525 < p < 10000$ (just hunt for lists of primes on the internet) and find a primitive root r of p. Don't do this just by googling, do this by making sure you understand what a primitive root is (what properties must it have?) and by sampling some possibilities (Wolfram Alpha can help) until you find one. Make sure you explain the process you followed so it's clear to the grader how you tested and validated. If you're not sure how to do this, ask!

   (b) Choose some a with $0 \leq a \leq p - 1$ and find the least nonnegative residue $b \equiv r^a \mod p$. What is your public key?

   (c) Create a message of your choosing with between 15 and 20 characters and encrypt it. Show as much work as we do in class, the actual calculations can be done elsewhere.