CS4286: Internet Security & E-commerce Protocols

Lecture 01: Admin & Introductory Security Concepts

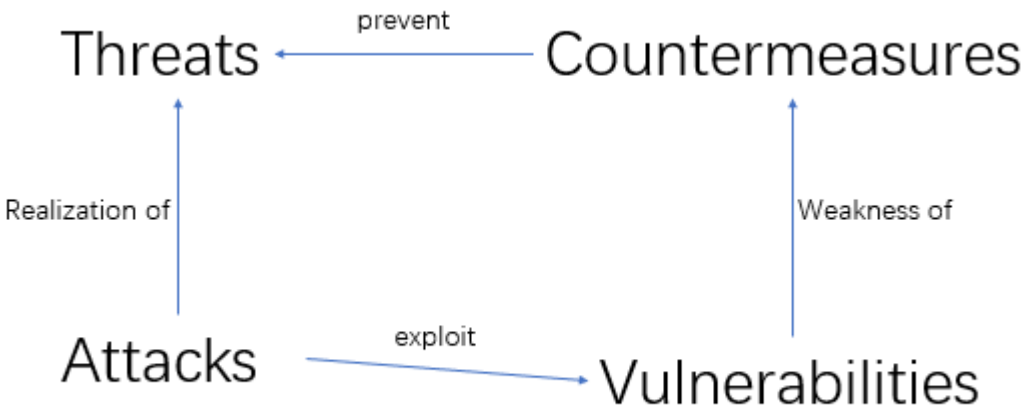
Intro

- Assessment:
  - 30 course work = 20 (assignments * 3) + 10 (midterm)
  - 70 final

Basic Concepts & Terminology

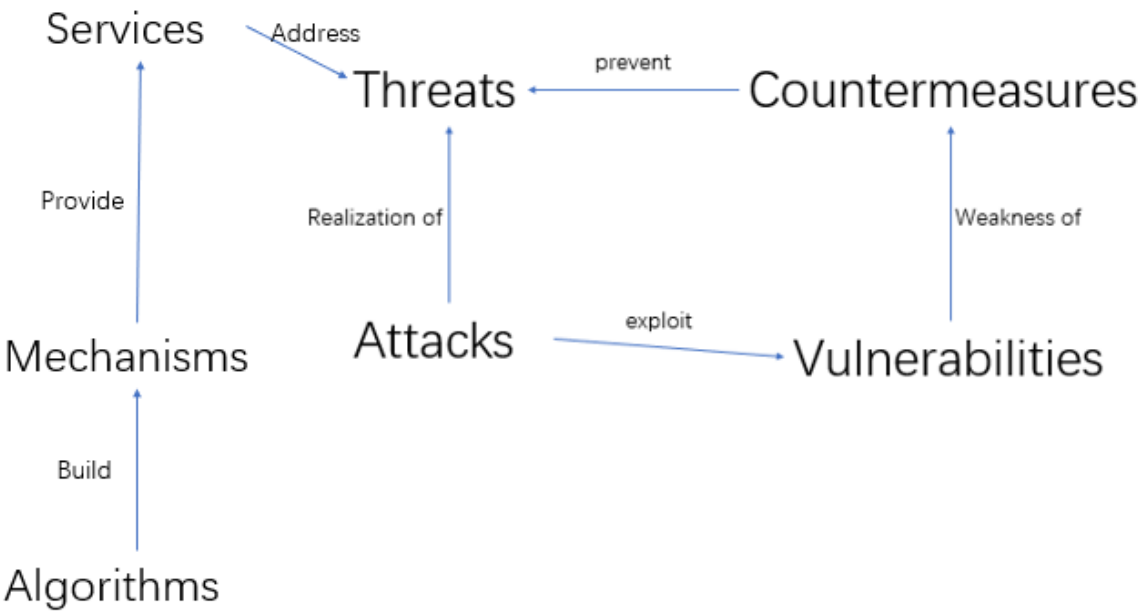
- What is Security: The security of a system, application, or protocol is always relative to (designed based on):
  - A set of **desired properties**: what do we want to achieve (whether the property is valuable)?
  - An **adversary**(attackers) with specific capabilities: what can they do?
  - What security (service) do we need?
  - How powerful an attacker do we want to defend against?
- Why important
  - Is good security always secure?
  - We need to think: Appropriate? Strength? Cost?
  - Unconditionally VS. Computationally
    - **Unconditionally(theoretically)**: Infinite resource cannot break
    - **Computationally**: Available resource cannot break (RSA)
    - **Financially**: Not profitable to break
- How to know security or not
  - single module
  - system
  - Majority of problems in real life secure systems is not directly due to weak crypto – but rather crypto used in wrong way (without considering context) or non-crypto issues (mostly human) that weaken security.
- Information Security:
  - Security is about the protection of assets.
  - Thus, **information security** is the basis for protecting our **information assets**
- Three broad classes of **protection measures**:
  - **Prevention**: prevent your assets from being damaged
  - **Detection**: detect when you assets have been damaged, by whom and how
  - **Reaction/Recovery**: recover your assets. or recover from the damage to your assets
  - Difference between prevention and detection: For example, we prevent the attacker from getting plaintext if we encrypt well but we cannot prevent modification during transmission (we can however detect it).

- Basic **Security Goals**: How can our information assets be compromised(危及)? => three **aspects** of information protection (CIA):
    - **Confidentiality(机密的)**: prevention of unauthorized **disclosure** of information.
    - **Integrity(正确的)**: prevention of unauthorized **modification** of information.
    - **Availability(可靠性)**: prevention of unauthorized **withholding** of information or resources. (refute to give)
    - CIA are most basic security goals => but there are more than three services
- Threats
    - Relationships



        - Security is only desirable when there is a need to protect a system from a threat.
        - A security threat is a **possible means by which a security policy may be breached** (e.g. loss of integrity or confidentiality).
        - Countermeasures are **controls to protect against threats**.
        - Vulnerabilities are **weaknesses** in the system (and/or **countermeasures**).
        - An attack is a **realization** of a **threat** (**exploiting** a **vulnerability**).
    - threats can be classified as:
        - deliberate (e.g. hacker penetration);
        - accidental (e.g. a sensitive file being sent to the wrong address).
    - The associated threats which **CIA** are responsible for countering are:
        - **Exposure of data**: the threat that someone who is unauthorized can access the data.
        - **Tampering with data**: the threat that the data could be altered from what it should be.
        - **Denial of service**: the threat that the data or service is unavailable when it is required.
        - For the three threats which is accidental and which deliberate? All three can be both.
- Adversaries(not always third party or outsiders):
    - People whose aim it is to circumvent your security are generally called adversaries.
        - Sometimes called intruders, but not all adversaries are external to the system (insider threats).
    - Adversaries act in two different ways:
        - **Passive adversaries** only attempt to get unauthorised access to information
        - **Active adversaries** take more direct action:
            - Unauthorized alteration
            - Unauthorized deletion
            - Unauthorised transmission

        - Falsification of origin of information
        - unauthorized prevention of access to information
    - you should not memorize what active adversaries can do, rather just think anyone that is not passive is active (not just access is active)
    - Think:
        - Someone reads your email – Active
        - Someone sends email to your friend pretending to be you – Passive
- Service & Mechanism



    - **Security Service**(High level security goal): A measure which can be put in place to address a threat(e.g. provision of confidentiality)
    - **Security Mechanism**(actual ways): A means to provide a service
- Services
    - Data Confidentiality and Integrity
        - Confidentiality: Protection against unauthorized disclosure of information. => focus on prevent in the three protection measures
            - What is information? Is it only data?
                - Is traffic flow confidentiality important? Lets say you encrypt the data but people can see when we sent it? - Decide what do you wish to protect and keep confidential.
                - Even if we make the ‘data’ confidential - we make the application layer payload confidential - that means the packet on the wire still has transport layer data (ports, sequences), network layer (destination address) and data link layer information.
                - Example, are encrypting routers in SWIFT networks (for secure financial messaging) – complete traffic confidentiality (cannot see destination) decrypt – look at network routing information, encrypt, send on.
        - Integrity: Integrity is protection against unauthorized modification of data => focus on detect, recover in the three protection measures, cannot prevent
            - includes corruption, deletion, addition and other modifications
    - Authentication

        - Entity authentication: Whether a specific person (message from which person)
            - Provides checking of a claimed identity at a point in time
            - Typically used at start of a connection.
            - Addresses masquerade(假扮) and replay threats
            - Come in person
        - Origin authentication: who did that (does not need provide entity authentication)
            - Provides verification of source of data
            - Does not protect against replay or delay
            - Signature of check, does not check the writer of the signature
    - Access Control:
        - Provides protection against unauthorized use of resource, including:
            - use of a **communications** resource
            - **reading, writing or deletion** of an information resource,
            - **execution** of a processing resource.
        - Subject action resource -- Is this person allowed to perform this action on this resource.
    - Non-repudiation: no denial
        - **non-repudiation of origin**: cannot say not send
            - Most people talk about origin – and this is more common than you think.
        - **non-repudiation of delivery**: cannot say not receive
- Service (Threats)
    - Confidentiality (data disclosure)
    - Integrity (data alteration)
    - Availability (DoS - deny of service)
    - Entity Authentication (masquerade)
    - Origin Authentication (forgery 伪造)
    - Non-repudiation (repudiation – it did not happen!)
    - Access Control (illegitimate access)
- Mechanisms:
    - A security mechanism is a means to provide a service.
    - Can be divided into two classes:
        - Specific security mechanisms, used to provide **specific security services**, e.g. digital signature
            - An example of a specific mechanism is a digital signature scheme. It can be used to provide several different services – it can obviously be used to give a data integrity service, but it can also be used to provide an origin or entity authentication service even when data integrity is not required. Similarly there are other ways to provide a data integrity service which don't require digital signatures (e.g. MACs or routing controls).
        - Pervasive security mechanisms, **not** specific to **particular** services, e.g. event detection, labelling.
            - An example of a pervasive mechanism is an event detection mechanism. It doesn't actively provide any security service. (You can't say that this data is transmitted confidentially because of the event detection mechanism.) However, it supports every security service by providing a way to detect compromises which may render specific security mechanism ineffective.
- Service (Mechanisms)

    - Confidentiality (encryption)
    - Integrity (MAC/digital signature)
    - Availability (redundancy)
    - Entity Authentication (authentication protocol)
    - Origin Authentication(MAC/digital signature)
    - Non-repudiation (digital signature)
    - Access Control (Access control model)
- Mechanisms (Algorithm)
    - Algorithms are used to build mechanisms
    - Example of mechanisms/algorithms:
        - Encryption: DES/3DES/AES (modes) or RSA/ECC
            - CAST(Canada), MISTY1/Camellia (Japan), SEED (Korea)
        - MAC: CBC mode, HMAC
        - Digital Signature: RSA, DSA, ECC
        - Hash: SHA-3
        - Random number: True or Pseudorandom

Standards

- Why standards? - They ensure any business offering products, services or processes is
    - cost-effective and time efficient: You do not have to develop the solution yourself – take time, take experts and this comes up with an ad-hoc solution that is likely to be as good.
    - commercially viable (可行的): customer feels more confident in your solution. The products appears more credible to the customer because our company is using best practice. Could open up a new market
    - credible (可信的): ties into above.
    - safe: Because you are taking best practice into account, and using a solution developed by experts the chances are our solution is safer?
    - Interoperability (互操作性): Could tie into commercial viability – standards allow different entities to produce components that work together (without these entities having a relationship).
- How to use standards?
    - Three common ways to use a standard
        - Use as the basis for new design (use the parts you need) - starting point of a new solution => risky but some standards are meant to be starting points
        - Certification is when a neutral third-party attests(证明) to a claim of compliance (合规) => Good but costs money and takes time
        - Compliance may be declared without recourse to third party certification
    - Most security standards do not require certification
        - Most of the security standards fall into one of two broad categories: they are either
            - standards which define **definite algorithms** (in which compliance can be easily checked and certification is somewhat of an expensive luxury)
            - or they provide guidance on how to **produce a system or service** (in which case they are advice and certification is not really intended).
- Why not standards(disadvantages)

    - Consensus decisions imply **compromise**.
    - Documents can be **inconsistently implemented**.
    - Commercial pressure can lead to **partial implementation**.
    - Aggressive market strategies by companies who adapt or extend standards can undermine their usefulness.
- Categories of standards
    - International standards: The main function of these bodies is the production of ‘base standards’.
    - North American standards: have become particularly important in security
    - Internet standards: loose collaboration between government, industry and academia
        - Internet Standards are quite odd things. Why is it hard to produce reliable Internet standards?
            - Strangely - products tend to be interoperable – large companies works together, almost like a standards body – your product needs to work with theirs. Defacto standard –> interoperability.
            - Big problem – once a standard is being used it is used by everyone. Once again if there is a standard and something goes wrong with it is difficult to correct.
    - Company Standards: also sometimes issue defacto standards for techniques that have been patented.
        - Any reasons these would not be seen as standards? - As per previous definition:
            - Standard body?
            - Consensus?
            - Are the completely unbiased?
- Standards bodies
    - **official** standards bodies: defining standards at worldwide or national level
    - **companies** who make standards: often with commercial interest – some good some bad, and some you have to use like payment card standards
    - Internet standards: which no on really looks after, simply technology that grows in usage, and once everyone uses it might be organized and published as a standard by IETS

Basic Number Theory & CRT & Euler Function & RSA

Basic Number Theory

- Modular Arithmetic:
    - Addition / Subtraction: $(x+y)\ mod\ n = ((x\ mod\ n) + (y\ mod\ n))\ mod\ n$
    - Multiplication:$
        - proof:

$$
\begin{gathered}
x = k_1 n + b_1, y = k_2 n + b_2 (k_1, k_2, b_1, b_2\ \in \mathbb{N}^*) \\
(xy)\ mod\ n = (k_1 n + b_1)(k_2 n + b_2)\ mod\ n \\
= (k_1 k_2 n^2 + (k_1 b_2 + k_2 b_1) n + b_1 b_2)\ mod\ n \\
= b_1 b_2 \\
(x\ mod\ n)(y\ mod\ n) = ((k_1 n + b_1)\ mod\ n)((k_2 n + b_2)\ mod\ n) \\
= b_1 b_2
\end{gathered}
$$

        - $x^y\ mod\ n = (x\ mod\ n)^y\ mod\ n$

- Example: find the last digit of 2^{100} in decimal:

$$
\begin{aligned}
2^3 \bmod 10 &= 8 \\
2^6 \bmod 10 &= 8 \times 8 \bmod 10 \to 4 \\
2^{12} \bmod 10 &= 4 \times 4 \bmod 10 \to 6 \\
2^{24} \bmod 10 &= 6 \times 6 \bmod 10 \to 6 \\
2^{48} \bmod 10 &= 6 \times 6 \bmod 10 \to 6 \\
2^{96} \bmod 10 &= 6 \times 6 \bmod 10 \to 6 \\
2^{100} \bmod 10 &= 2^{96} \times 2^3 \times 2^1 \bmod 10 \to 6
\end{aligned}
$$

  - Note that if $ac \equiv bc (mod\ n)$ we can get $a \equiv b (mod\ n)$ only when $gcd(n, c) = 1$, or generally:

$$a \equiv b (mod\ \frac{n}{gcd(c, n)})$$

- Greatest Common Divisor(GCD)
  - Definition: The largest divisor shared by a given pair of integers
  - Coprime: Two integers x, y are coprime(relatively prime) iff $gcd(x, y) = 1$
  - Euclidean's Algorithm: If $a = qb + r$ for some integer q and r, then:

$$gcd(a, b) = gcd(r, b) = gcd(a\ mod\ b, b)$$

    - proof:

$$
\begin{gathered}
r = a\ mod\ b \\
r = a - qb \\
\frac{r}{d} = \frac{a}{d} - \frac{qb}{d} = m \\
(where\ d\ is\ a\ common\ divisor\ of\ a\ and\ b => m \in \mathbb{N}^*) \\
m\ is\ a\ divisor\ of\ r
\end{gathered}
$$

    - code:

```
int gcd(int a, int b)
{
    // Everything divides 0
    if (a == 0)
       return b;
    if (b == 0)
       return a;

    // base case
    if (a == b)
        return a;

    // a is greater
```

```
    if (a > b)
        return gcd(a%b, b);
    return gcd(a, b%a);
}
```

- Extended Euclidean's Algorithm:
  - problem: Solve the equation $ax + by = gcd(a, b)$
    - Always keep the first parameter of gcd to be the larger one
    - The equation always has a solution
  - Base case: $b = 0$, that means the previous $gcd(a, b)$ satisfies $a\%b = 0$, and b becomes current a, then a itself is gcd => $gcd * 1 + 0 = gcd$
  - Other cases: $b \neq 0$: according to the Euclidean's Algorithm $gcd(a, b) = gcd(b, a\ mod\ b)$

$$ax + by = gcd(a, b), a = bq + r, r = a - floor(\frac{a}{b})b$$
$$gcd(b, a\ mod\ b) = gcd(b, r) = bx' + ry' = bx' + (a - floor(\frac{a}{b})b)y'$$
$$= bx' + ay' - floor(\frac{a}{b})by'$$
$$= ay' + b(x' - floor(\frac{a}{b})y')$$
$$gcd(a, b) = gcd(b, r)$$
$$x = y', y = x' - floor(\frac{a}{b})y'$$

  - code:

```
int exGcd(int a,int b,int &x,int &y)
{
    if(b==0)
    {
        x=1;y=0;
        return a;
    }
    int r=exGcd(b,a%b,x,y);
    int t=x;x=y;y=t-a/b*y;
    return r;
}
```

- Least Common Multiple (LCM):

$$lcm(x, y) \geq max(x, y)\ for\ any\ x, y$$
$$lcm(x, y) \leq xy$$
$$lcm(x, y) = \frac{xy}{gcd(x, y)}$$

- Modular Inverse:
  - Definition: b^{-1} is called a inverse of b iff

$$bb^{-1} \equiv 1(mod\ n)$$

  - The inverse doesn't always exist (e.g. $2x \equiv 1(mod\ 4)$)
  - Inverse exists only if

$$gcd(b,n)=1$$

- Proof:
  - When $gcd=1$

$$bs+nt=gcd(b,n)=1$$
$$(bs+nt)\ mod\ n=bs\ mod\ n=1$$
$$s=b^{-1}$$

    - s,t are integers, gcd is less than or equal to minimum of b and n, so s and t must have one negative number (otherwise, the value of gcd will be larger than either b or n, which is a contradiction)
  - If gcd is not 1, the minus(difference) between multiple of b and n will be multiple of $gcd(b,n)$:

$$Assume\ b=gcd(b,n)*k_1, n=gcd(b,n)*k_2$$
$$bs+nt=k_1s*gcd(b,n)+k_2t*gcd(b,n)$$
$$=gcd(b,n)*(k_1s+k_2t)$$

Chinese Reminder Theorem (CRT)

- Problem: Find the value of x such that:

$$\begin{cases} x\equiv a_1\ (mod\ n_1) \\ x\equiv a_2\ (mod\ n_2) \\ \vdots \\ x\equiv a_k\ (mod\ n_k) \end{cases}$$

-

Euler Function and Euler Theorem

- Definition: Euler Function ϕ is defined as:

$$\phi(n)=|\{x\in(\mathbb{N}^*\cap[1,n-1])\mid gcd(x,n)=1\}|\ where\ (n\in(\mathbb{N}^*\cap[1,+\infty))$$

  - The number of positive integers less than n such that n and the integers are coprime.
  - Denote the set:

$$Z_n=\{x\in(\mathbb{N}^*\cap[1,n-1])\mid gcd(x,n)=1\}\ where\ (n\in(\mathbb{N}^*\cap[1,+\infty))$$
$$|Z_n|=\phi(n)$$

- The value of Euler Function:
  - For prime number p:

$$\phi(p)=p-1$$

  - For prime number p , q:

$$\phi(pq)=(pq-1)-(p-1)-(q-1)=pq-p-q+1=(p-1)(q-1)$$

    - proof:

$$gcd(p,q)=1$$
$$Z_{pq}=\{1,2,\dots,pq-1\}-\{p,2p,\dots,(q-1)p\}-\{q,2q,\dots,(p-1)q\}$$
$$|Z_{pq}|=(pq-1)-(p-1)-(q-1)$$

- For prime number p and a positive integer k:

$$\phi(p^k) = p^k - p^{k-1} = p^k(1 - \frac{1}{p})$$

  - proof:
    - p^k is not calculated in the set, $p^k - p$ is the last multiple of p

$$\begin{aligned} Z_{p^k} &= \{1, 2, \dots, p^k - 1\} - \{p, 2p, \dots, (p^{k-1} - 1)p\} \\ &= (p^k - 1) - (p^{k-1} - 1) \end{aligned}$$

- For positive integer p, q, $gcd(p, q) = 1$

$$\phi(pq) = \phi(p) * \phi(q)$$

  - proof: we only need to prove there is a double mapping from cartesian product of Z_p and Z_q to Z_{pq}
    - From left to right: prove $a \in Z_p, b \in Z_q \longrightarrow (aq + bp)\ mod\ (pq) \in Z_{pq}$
      - Prove the correspondence: Euclidean Algorithm is used in the last two steps

$$\begin{gathered} gcd(a, p) = gcd(b, q) = gcd(p, q) = 1 \\ (aq + bp)\ mod\ p = aq, (aq + bp)\ mod\ q = bp \\ gcd(aq + bp, p) = gcd(p, aq) = 1 = gcd(q, bp) = gcd(aq + bp, q) \\ Therefore,\ gcd(aq + bp, pq) = 1 \longleftrightarrow gcd((aq + bp)\ mod\ pq, pq) = 1 \\ (aq + bp)\ mod\ pq \in Z_{pq} \end{gathered}$$

      - Prove no two pairs $(a_1, b_1), (a_2, b_2)$ in $Z_p \times Z_q$ corresponds to the same value in Z_{pq} (one-to-one)

$$\begin{gathered} (a_1 - a_2)q + (b_1 - b_2)p = kpq \\ (b_1 - b_2)p \equiv 0\ (mod\ q) \\ b_1 \equiv b_2 (mod\ q) \\ b_1 < q, b_2 < q \\ b_1 = b_2 \\ similarly, a_1 = a_2 \end{gathered}$$

    - From right to left: Define function $f : Z_{pq} \mapsto Z_p \times Z_q\ s.t.\ f(x) = (x\ mod\ p, x\ mod\ q)$
      - Prove the correspondence:

$$\begin{gathered} gcd(pq, x) = 1 \\ gcd(p, x) = 1, gcd(q, x) = 1 \\ gcd(x\ mod\ p, p) = gcd(x\ mod\ q, q) = 1 \\ x\ mod\ p \in Z_p, x\ mod\ q \in Z_q \end{gathered}$$

      - One-to-One

$$\begin{gathered} Assume\ a, b \in Z_{pq}, a \neq b, f(a) = f(b) \\ a \equiv b\ (mod\ p), a \equiv b\ (mod\ q) \\ gcd(p, q) = 1 \longrightarrow a \equiv b\ (mod\ pq) \\ a, b \in [1, pq) \cap \mathbb{N}^* \\ a = b\ (contradiction) \end{gathered}$$

- For any positive integer n, which is a multiple of r prime numbers:

$$n = p_1^{k_1} p_2^{k_2} \dots p_r^{k_r} (k_i \geq 1)$$
$$\phi(n) = \phi(p_1^{k_1})\phi(p_2^{k_2}) \dots \phi(p_n^{k_n})$$
$$= p_1^{k_1} p_2^{k_2} \dots p_r^{k_r} (1 - \frac{1}{p_1})(1 - \frac{1}{p_2}) \dots (1 - \frac{1}{p_r})$$
$$= \Pi_{i=1}^{r} p_i^{k_i} (1 - \frac{1}{p_i})$$
$$= n \, \Pi_{i=1}^{r} (1 - \frac{1}{p_i})$$

- Euler's Theorem
  - For positive integer a and n, if $gcd(a, n) = 1$, $a^{\phi(n)} \equiv 1 (mod\ n)$ ($a^{-1} \equiv a^{\phi(n)-1}\ (mod\ n)$)
  - proof:
    - $Z_n = \{x_1, x_2, \dots, x_{\phi(n)}\}, S = \{(a * x_1)\ mod\ n, (a * x_2)\ mod\ n, \dots, (a * x_{\phi(n)})\ mod\ n\}$, then $Z_n = S$

$$gcd(x_i, n) = gcd(a, n) = 1$$
$$gcd(ax_i, n) = 1, gcd(ax_i\ mod\ n, n) = 1$$
$$ax_i\ mod\ n \in Z_n$$
$$Assume\ ax_i \equiv ax_j (mod\ n)$$
$$a(x_i - x_j) = kn, gcd(a, n) = 1$$
$$x_i - x_j \equiv 0 (mod\ n), x_i < n, x_j < n$$
$$x_i = x_j$$

    - According to the equivalent relationship above:

$$(a^{\phi(n)} * \Pi_{i=1}^{\phi(n)} x_i)\ mod\ n$$
$$= (ax_1\ mod\ n) * (ax_2\ mod\ n) * \dots (ax_{\phi(n)}\ mod\ n)\ mod\ n$$
$$= \Pi_{s \in S} s = \Pi_{x \in Z_n} x$$
$$= (x_1 * x_2 * \dots * x_n) mod\ n = (\Pi_{i=1}^{\phi(n)} x_i) mod\ n$$
$$a^{\phi(n)} \equiv 1 (mod\ n)$$

RSA Algorithm

- Notations:
  - M: message
  - C: encrypted message
  - $\{n, e\}$: public key
  - $\{d, n\}$: private key
- Set up:
  - pick two prime numbers (larger better) p, q
  - $n = p * q$, the length of n_{binary} is the length of the encrypt key
    - we require that $M < n$ during encryption
  - compute $\phi(n) = (p - 1)(q - 1)$
  - pick an integer e **randomly** such that $gcd(e, \phi(n)) = 1$
  - compute $d \equiv e^{-1} (mod\ \phi(n))$

$$ed \equiv 1(mod\ \phi(n))$$
$$ed - 1 = k\phi(n)$$
$$ex + \phi(n)y = 1$$

- Encryption:

$$C = M^e\ mod\ n (where\ M < n)$$

- Decryption:

$$M = C^d\ mod\ n = M^{ed}\ mod\ n$$

- Proof of correctness:

$$ed \equiv 1(mod\ \phi(n))$$
$$ed = h\phi(n) + 1$$
$$M^{h\phi(n)+1}\ mod\ n = M$$

  - when $gcd(M, n) = 1$, according to Euler's theorem

$$M^{h\phi(n)} \equiv 1(mod\ n)$$
$$M^{h\phi(n)+1} \equiv M(mod\ n)$$

  - when $gcd(M, n) \neq 1$, according to Euler's theorem
    - By assumption, $M < n$

$$n = pq \leftrightarrow M = kp\ or\ M = kq$$
$$if\ gcd(k, q) \neq 1, q\ is\ prime, M = kp > pq = n$$
$$(kp)^{q-1} \equiv 1(mod\ q)$$
$$[(kp)^{q-1}]^{h(p-1)} * kp \equiv kp(mod\ q)$$
$$(kp)^{h\phi(n)+1} \equiv kp(mod\ q)$$
$$(kp)^{ed} \equiv kp(mod\ q)$$
$$(kp)^{ed} = tq + kp$$
$$tq = kp[(kp)^{ed-1} - 1]$$
$$gcd(p, q) = 1 \rightarrow t = t'p$$
$$(kp)^{ed} = t'pq + kp$$
$$M^{ed} \equiv M(mod\ n)$$

- Security
  - is it possible to calculate d with known n and e

$$ed \equiv 1(mod\ \phi(n))(need\ \phi(n))$$
$$\phi(n) = (p-1)(q-1)\ (need\ p\ q)$$
$$n = pq$$

  - if n can be break into two prime numbers, d can be computed => harder to break large n, harder to break RSA