Digital Signatures

Cole Watson

Opening Activity

- Dr. Cusack owns a lockbox, padlock, and keys.
- The padlock is private and unique to him.
- The keys are public and they can only open Dr. Cusack's lockbox.
- Only Dr. Cusack can lock his padlock (It's a very smart padlock).

Activity (cont)

- Dr. Cusack has promised the whole class that everyone will receive an A on their final exam.

- To hold true to his word, he writes his promise on a piece of paper and locks it into the lockbox with his padlock.

- He then gives his keys to President Knapp because he is a trusted source.

Activity (cont)

- Flash forward to the day after the final exam.
- Dr. Cusack grades all the final exams using the stair method and no one receives an A.
- The class is outraged. Dr. Cusack has lied!
- Everyone then decides to go to the Provost to make sure that they all get the A that Dr. Cusack promised them.

Activity (cont)

- The Provost hears what the students have to say and he asks them to prove their claim.

- To do this they grab Dr. Cusack's lockbox and get Dr. Cusack's key from President Knapp and they unlock it.

- Inside holds the note that promises all the students an A on their final exam.

Activity (cont)

- The note can only be from Dr. Cusack since only his public key can unlock his unique padlock.

- Dr. Cusack, although reluctantly, gives all the students an A on their final exam.

RSA Digital Signature Formula

Let $n = pq$, where p and q are primes. Let $\mathcal{P} = \mathcal{A} = \mathbb{Z}_n$, and define

$$\mathcal{K} = \{(n, p, q, a, b) : n = pq, p, q \ prime, \ ab \equiv 1 \ (mod \phi(n))\}.$$

The values n and a are public, and the values $p, \ q, \ b$ are secret.

For $K = (n, p, q, a, b)$, define

$$sig_K(x) = x^b \ mod \ n$$

and

$$ver_K(x, y) = true \Leftrightarrow x \equiv y^a \ (mod \ n)$$

$(x, y) \in \mathbb{Z}$.

…ton: CRC
Press, 1995.

Proof of RSA Scheme

The formal description of the cryptosystem is given in Figure 4.2. Let's verify that encryption and decryption are inverse operations. Since

$$ab \equiv 1 \pmod{\phi(n)},$$

we have that

$$ab = t\phi(n) + 1$$

for some integer $t \geq 1$. Suppose that $x \in \mathbb{Z}_n{}^*$; then we have

$$\begin{aligned}(x^b)^a &\equiv x^{t\phi(n)+1} \pmod n \\ &\equiv (x^{\phi(n)})^t x \pmod n \\ &\equiv 1^t x \pmod n \\ &\equiv x \pmod n,\end{aligned}$$

Taken from: Stinson, Douglas R. *Cryptography: Theory and Practice*. Boca Raton: CRC Press, 1995.

Why does $x^{\phi(n)} = 1$?

- $\phi(n)$ is the Euler function, which is defined as the number of positive integers that are relatively prime to n.
- The group of units, U(n) is defined as the elements in $\mathbb{Z}_n$ that are relatively prime to n.
- The order of U(n) is $\phi(n)$.
- Thus when $x \in U(n), x^{\phi(n)} = 1$.
- When $x \notin U(n)$ there is a more complicated proof, but the result is the same.

Why does $\phi(pq) = (p-1)(n-1)$?

3a) *Proof*: By Corollary 4.7, the generators of $\mathbb{Z}_{pq}$ are all the integers r such that $1 \leq r < n$ and $gcd(r, pq) = 1$. Thus the numbers $p, 2p, 3p, ..., (q-1)p$ cannot generate $\mathbb{Z}_{pq}$ since they are all multiples of p. Similarly, the numbers $q, 2q, 3q, ..., (p-1)q$ cannot generate $\mathbb{Z}_{pq}$ since they are all multiples of q. It is clear that 0 also cannot generate $\mathbb{Z}_{pq}$. Then the total number of generators of $\mathbb{Z}_{pq}$ is $pq - (q-1) - (p-1) - 1 = pq - q - p + 2 - 1 = pq - q - p + 1 = p(q-1) - (q-1) = (p-1)(q-1)$. Therefore there are (p-1)(q-1) generators of $\mathbb{Z}_{pq}$□

Proof written by me in my Abstract Algebra class

Euclidean Algorithm Example

- gcd(81,57)
- $81 = 1(57) + 24$
- $57 = 2(24) + 9$
- $24 = 2(9) + 6$
- $9 = 1(6) + 3$
- $6 = 2(3) + 0.$

Finding the Inverse in $\mathbb{Z}_n$

- If gcd(a,b) = r, then there exist integers p and s such that p(a) + s(b) = r.
- x has an inverse if and only if gcd(x,n) = 1.
- Then p, and s exist such that px + sn = 1.
- px = 1 + (-s)n, so $px \equiv 1\ (mod\ n)$.
- To find p, we will use the extended Euclidean algorithm.

Example on Whiteboard

- Find inverse of 15 mod 26.

- Extended Euclidean Algorithm
  - $p_{i-2} - p_{i-1} \cdot q_{i-1} \text{ (mod n)}$
  - $p_0 = 0, p_1 = 1.$

In Class Worksheet

- Split into two groups.

Attacks on Digital Signatures

- No message attack
- Chosen message attack

No Message Attack

- Try to generate new valid signatures without the knowledge of the private key.
- Attacker obtains victims public verification key.
- Attacker finds a message x and a signature for x that can be verified with the victims public key.
- Called no message attack since no valid signatures from other documents are used.

No Message Attack (cont.)

- Oscar chooses an integer s between 0 and n.
- He claims that it is a signature of Alice.

- Bob wants to verify this signature so he uses Alice's public verification key to do this.

- If the message is meaningful text, then Oscar has successfully forged Alice's signature.

Chosen Message Attack

- Attacker knows valid signatures and uses them to create new signatures.
- Possible for an attacker to obtain signatures of their choosing.
- From two valid signatures, a third can be computed.

Chosen Message Attack

- Let m be a message. The attacker chooses an m_1 that is different than m, such that gcd(m, m_1) = 1.

- Calculates $m_2 = m{m_1}^{-1} \, mod \, n$

- Then the attacker uses the valid signatures s_1, s_2, for m, m_1 to compute $s = s_1 s_2$ mod n.

Cryptographic Hash Functions

- Map strings of an arbitrary length to a fixed length string of size between 128 and 512 bits.

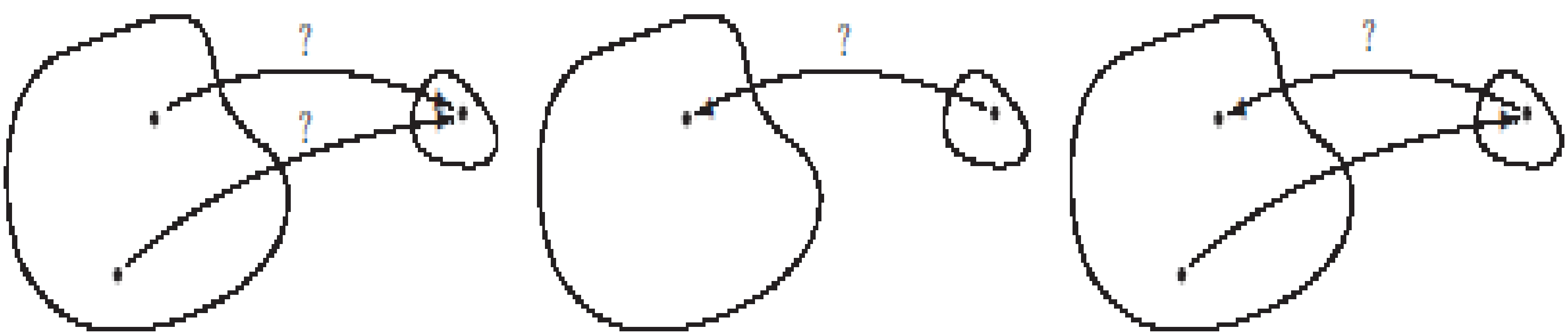
- Always expected to be one way.
  - Given a message y in the image, it is practically impossible to find a message x such that H(x) = y.

- Each message should have a different hash value.
  - This usually is not true, but it should be almost impossible to find two messages with the same hash value.

Hash Function Properties

- Collision resistance
  - Difficult to find two messages that hash to the same value.
- Preimage resistance
  - Given hash value of a message, it should be difficult to find any message hashing to that value.
- Second preimage resistance
  - Given some message, it should be difficult to find a different message that has the same hash value.

Properties (cont.)



(a) Collision (b) Preimage (c) Second preimage

FIgure taken from:
Thomsen, Søren Steffen. *Cryptographic Hash Functions*. Technical paper. Technical University of Denmark Department of Mathematics.

Signatures With Hash Functions

- Instead of computing the signature with the full document x, we compute the signature on the hash value of x.

- $s = h(x)^a \pmod n$.

Signatures With Hash Functions (cont.)

- To verify the signature we do:
  - $ver = s^b \ (mod\ n)$

- If $ver = h(x)$ then the signature is authentic.

- The hashing function is public and x is shared, thus it is easy to compute h(x).

Prevents Attacks

- No message attacks don't work since the attacker must come up with an x such that $h(x) = s^a \; mod \; n$.
  - Because the hash function is one way such an x cannot be computed.

- Chosen message attacks don't work since h is one way it is impossible to find x such that $h(x) = m = m_1 m_2 \; mod \; n$.

Public Key Infrastructures

- It is very important to keep private keys private and public keys safe from falsification.

- Thus a *personal security environment (PSE)* is needed.
  - Keys and securely stored here.
  - The signing and decrypting also done here to keep private keys secure.

Certification Authorities

- Each public key user is associated with a trusted *certification authority (CA)*.
- The CA certifies the correctness and validity of the public keys of it's users.
- The users know their CA's public key and can thus use it to verify the signatures from their CA.

Certification Authorities (cont.)

- Registration
  - Tell CA name and other personal info.
  - Present identification by going to CA in person.
  - Given a unique username.
- Key Generation
  - Generated in PSE or by CA.
  - Recommended that individuals don’t know their private keys.
  - Private keys are stored in PSE
  - Public keys in CA.

Certification and Archive

- Certification
  - CA generates certificate which establishes verifiable connection between user and public keys.
- Archive
  - Public key systems must be stored even after they expire.
  - CA stores certificates for public signature keys.

References

1. Buchmann, Johannes. *Introduction to Cryptography*. New York: Springer, 2004.

2. Engelfriet, Arnoud. "Crash Course on Cryptography: Digital Signatures." (in Technology Encryption Crash Course @ Iusmentis.com). October 1, 2005. http://www.iusmentis.com/technology/encryption/crashcourse/digitalsignatures/.

3. "Extended Euclidean Algorithm." Extended Euclidean Algorithm. Accessed December 08, 2015. http://www-math.ucdenver.edu/~wcherowi/courses/m5410/exeucalg.html.

4. Judson, Thomas W. *Abstract Algebra: Theory and Applications*. Boston, MA: PWS Pub., 1994.

5. Stinson, Douglas R. *Cryptography: Theory and Practice*. Boca Raton: CRC Press, 1995.

6. Thomsen, Søren Steffen. *Cryptographic Hash Functions*. Technical paper. Technical University of Denemark Department of Mathematics.