CSC3631 Cryptography
- Asymmetric Encryption I

Thomas Gross

1

Overview

Easy Land (Polynomial-Time)

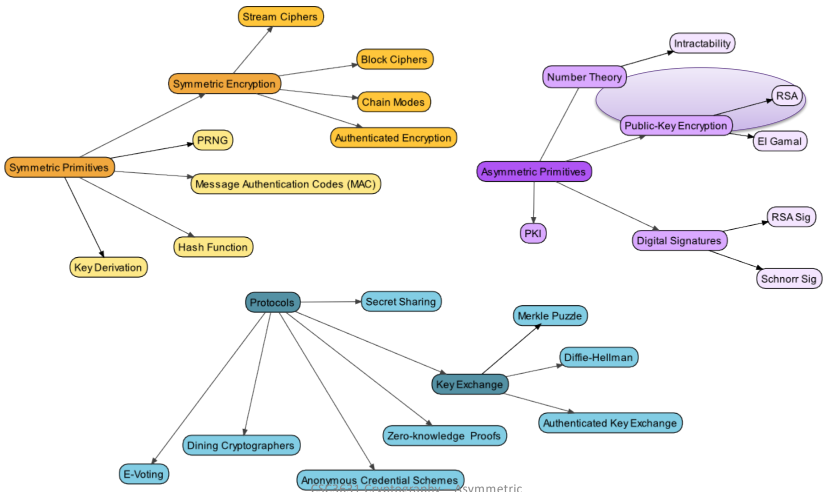
Addition:	a+b (mod N)	**Multiplication:**	a · b (mod N)
Inverse:	a^{-1} (mod N)	**Exponentiation:**	a^b (mod N)

Hard Land (Intractability)

Factoring What are p, q st. p · q = N?	**RSA** What's x st. $x^e = y$ (mod N)?	**Discrete Log** What's x st. $g^x = h$ (mod p)	**Diffie-Hellman** Distinguish g^{xy} from g^z
Integers	**RSA Group** $(Z_N)^*$, N=pq	**Subgroups of** $(Z_p)^*$	**Subgroups of** $(Z_p)^*$
Easy: Multiplication N = p · q	**Easy: Exponentiation** $x^e = y$ (mod N)	**Easy: Exponentiation** $g^x = h$ (mod p)	**Easy: Exponentiation** $K = g^{xy}$ (mod p)
Hard: Find factors of N	**Hard:** Find the e^{th} root x given y.	**Hard:** Find the discrete logarithm x given h.	**Hard:** Decide whether K is DH or random

21

Big Picture



22

Roadmap

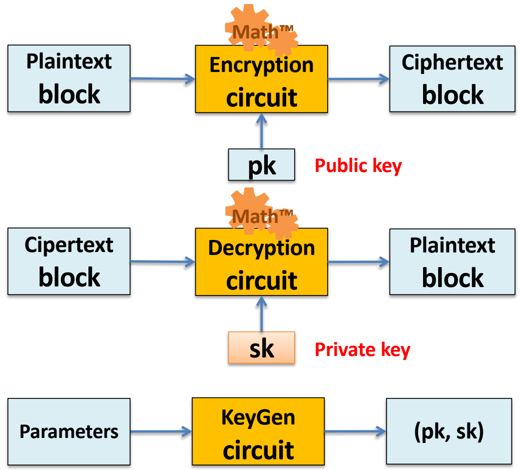
- **Asymmetric Encryption**
- Chosen-Plaintext Attack Security
- Hybrid Encryption
- RSA Encryption
  - Textbook RSA
  - RSA with Padding

Goal for today:

- **How to encrypt with asymmetric means?**
- **What are the nuts and bolts of RSA encryption?**

23

Structure



Characteristics

- **Goal:** confidentiality
- Keep private key sk absolutely secret.
- Public key pk is inadvertently **public**.
  (also known by the adversary)
- The private key sk **cannot** be deduced from pk.
- Publish pk by distributing it integerly.

Pros & Cons Asymmetric Encryption

Advantages	Disadvantages
No pre-shared secret	2-3 orders of magnitude slower than symmetric tools
Keys independent of sender	No data origin authentication or integrity
Anyone who wants to encrypt to Alice can do so.	Risk of **impersonation** attacks
Only a single private key to keep secret.	

Key Exchange w/ Public-Key Encryption

IMPORTANT for all public-key encryption:
Key distribution MUST be authentic. Otherwise, an active adversary can create an **impersonation attack.**

Key Distribution

Simplified

Each party only needs to distribute pk

Important

Key distribution must be guaranteed to done with integrity!

How?

Roadmap

- Asymmetric Encryption
- **Chosen-Plaintext Attack Security**
- Hybrid Encryption
- RSA Encryption
  - Textbook RSA
  - RSA with Padding

Goal for today:

- **How to encrypt with asymmetric means?**
- **What are the nuts and bolts of RSA encryption?**

Chosen-Plaintext Attack Security I

A secure cipher should produce ciphertext that is indistinguishable from random.

Written as a game with Adversary **A**.

Setup: Generate keypair (pk, sk)

Inputs to Adversary **A:** pk, $\mathbf{Enc}_{pk}()$

A produces candidate messages m_1, m_2

Choose random bit b ← {0,1}

Give challenge ciphertext to A: c ← $\mathbf{Enc}_{pk}(m_b)$

Outputs from **A:** a bit b'

Success criterion for **A:** if guess correct b'=b

Chosen-Plaintext Attack Security II

How can we formalize CPA indistinguishability?

Setup: Generate keypair (pk, sk)

pk, $\mathbf{Enc}_{pk}()$ →

← candidate messages m_1, m_2

Choose random bit b ← {0,1}

Challenge ciphertext c ← $\mathbf{Enc}_{pk}(m_b)$

c →

← a bit b'

Success criterion for **A:** if guess correct b'=b

Deterministic Public-Key Encryption

No deterministic encryption can be CPA-secure.
Public-key encryption must be randomized.

Example (encryption of short messages):

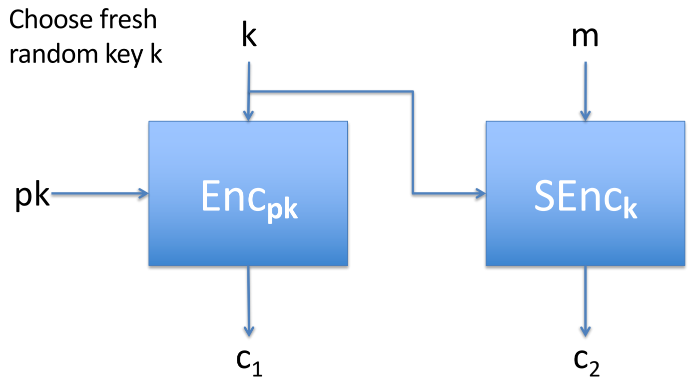
Roadmap

- Asymmetric Encryption
- Chosen-Plaintext Attack Security
- **Hybrid Encryption**
- RSA Encryption
  - Textbook RSA
  - RSA with Padding

Goal for today:

- **How to encrypt with asymmetric means?**
- **What are the nuts and bolts of RSA encryption?**

Hybrid Encryption



More efficient than pure asymmetric encryption.
Advantage of asymmetric enc: no pre-shared secret key.

Take-home Messages

- Key distribution must be **authentic**.
- Ciphertext **indistinguishable from random**.
- **Deterministic** public-key encryption **insecure**.
- Efficiency: Use **hybrid encryption**!

Roadmap

- Asymmetric Encryption
- Chosen-Plaintext Attack Security
- Hybrid Encryption
- **RSA Encryption**
  - **Textbook RSA**
  - RSA with Padding

Goal for today:

- **How to encrypt with asymmetric means?**
- **What are the nuts and bolts of RSA encryption?**

Recall Euler's Theorem

Euler

$x^{\varphi(N)} = 1$ (mod N)

$\mathbf{Z}_N$

RSA Crypto

Order of $(\mathbf{Z}_N)^*$

$\varphi(N) := |(\mathbf{Z}_N)^*|$

(Number of elements)

Knowing $\varphi(N)$

Gives us an advantage over an adversary.

The RSA Assumption

Recall: What's the basis of the RSA crypto system?

Setup: (N, e, d) ← **GenRSA**(1^n), where e · d =1 **mod** $\varphi(N)$

Choose y from $(\mathbf{Z}_N)^*$

Input for Adversary **A**: N, e, y

Output of Adversary **A**: x in $(\mathbf{Z}_N)^*$

Adversary A success: if x^e = y (**mod** N)

The RSA problem is **hard** relative to GenRSA if all probabilistic and polynomial-time adversaries **A** only have negligible success probability.

RSA Key Generation

How to create a strong setting for RSA?

GenRSA(1^n)

Input: key length n

Generate two large n-bit **distinct primes** p and q

Compute $N = p \cdot q$ and $\varphi(N)$ = (p-1) · (q-1)

Choose a random integer e, **gcd**(e, $\varphi(N)$) = 1

Compute e's inverse d: $d \cdot e = 1$ (**mod** $\varphi(N)$)

Output: $pk = (N, e)$, sk = (N, d)

Textbook RSA Encryption

KeyGen: $pk=(N, e)$, $sk=(N, d) \leftarrow \mathbf{GenRSA}(1^n)$

Enc: Given $pk=(N, e)$ and message m:

$$c = m^e \quad (\mathbf{mod}\ N)$$

Dec: Given $sk=(d, N)$ and ciphertext c:

$$m = c^d \quad (\mathbf{mod}\ N)$$

Correctness

Need to show:

$$\mathbf{Dec}_{sk}(\mathbf{Enc}_{pk}(m)) = m$$

Key: $\mathbf{gcd}(e, \varphi(N)) = 1$ and $ed = 1\ (\mathbf{mod}\ \varphi(N))$

$$c^d = (m^e)^d = m^{de \,(\mathrm{mod}\ \varphi(N))} = m^{(1+k\varphi(N))} = m^1\, m^{k\varphi(N)} = m$$

RSA Example

Primes: $p = 2357$, $q = 2551$

Composite modulus: $N = p \cdot q = 6012707$

$\varphi(N) = 6007800$

Choose e: 3674911 Find d: 422191

$e \cdot d = 1\ (\mathbf{mod}\ \varphi(N))$: $3674911 \cdot 422191\ (\mathbf{mod}\ 6007800)$

m = 5234673

$c = m^e\ (\mathbf{mod}\ N) = 5234673^{3674911}\ (\mathbf{mod}\ 6012707)$

$= 3650502$

[Example from Menezes et al., Handbook of Applied Cryptography]

How Secure is Textbook RSA?

- Textbook RSA is CPA-secure against active adversaries under the RSA assumption.
- Textbook RSA is CPA-secure against eavesdroppers under the RSA assumption.
- Textbook RSA is not secure at all, even if the RSA assumption holds.
- Textbook RSA is not secure at all and not even a proper encryption.

RSA as Pseudo-Random Trapdoor Permutation

GenRSA(1^n) provides $pk=(N, e)$, $sk=(N, d)$

Permutation $(\mathbf{Z}_N)^* \rightarrow (\mathbf{Z}_N)^*$**:**

$$y = x^e \ (\textbf{mod}\ N)$$

Reverse lookup with trapdoor d:

$$y^d = x \ (\textbf{mod}\ N)$$

Currently only known method to compute the e^{th} root: **factoring N**, but no reduction is known and there is evidence that none exists.

Encrypting with small e

Assume e chosen as 3

For small m, there's trouble. What can go wrong?

Common Modulus Attack

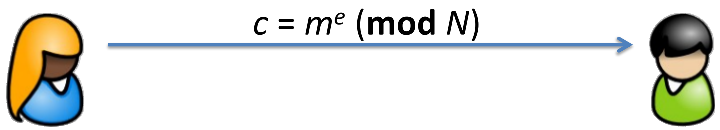
Assume organization uses **common modulus** N for all employees.

Each employee receives key pair ($pk=e$, $sk=d$)

What can go wrong?

Mangling Ciphertexts

Example: Alice sends bid m=**1000** in an auction.



Small Decryption Exponent

Small decryption exponent $d < N^{0.3}$

One can compute d from e and N

Choose decryption exponent d large enough: $d > N^{1/2}$

Roadmap

- Asymmetric Encryption
- Chosen-Plaintext Attack Security
- Hybrid Encryption
- **RSA Encryption**
  - Textbook RSA
  - **RSA with Padding**

Goal for today:

- **How to encrypt with asymmetric means?**
- **What are the nuts and bolts of RSA encryption?**

RSA with PKCS #1 v1.5 Padding

Idea: Prefix D-byte message m with random padding

Encryption:

Choose random byte-string r (k-D-3>8 bytes).

16 bit		8 bit	
2	Random padding r	0	m

$(00000000||00000010||r||00000000||m)^e$ (**mod** N)

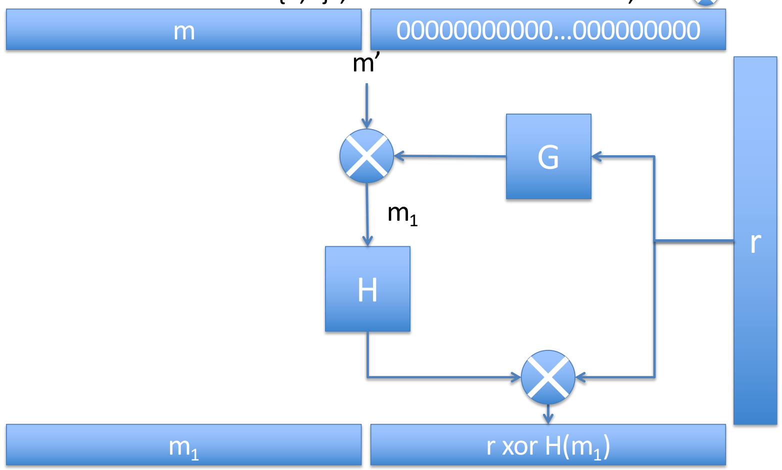
Decryption:

As usual, check that the padding is ok!

Believed to be a **CPA-secure encryption**, but no proof for that exists.

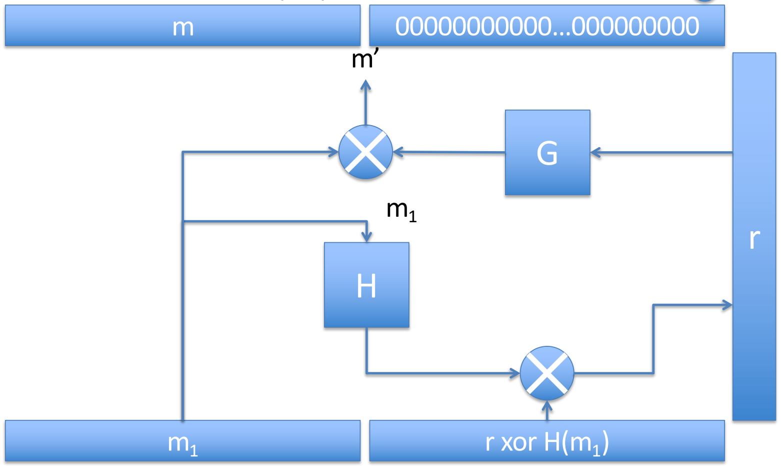
RSA OAEP Encryption

Choose random r in $\{0,1\}^n$; Hash-Functions **G** and **H**; XOR



RSA OAEP Decryption

Choose random r in $\{0,1\}^n$; Hash-Functions **G** and **H**; XOR $\otimes$



RSA in Practice: Hybrid with padding

Putting it all together

Choose fresh random x in $(\mathbf{Z}_N)^*$