Lecture 11

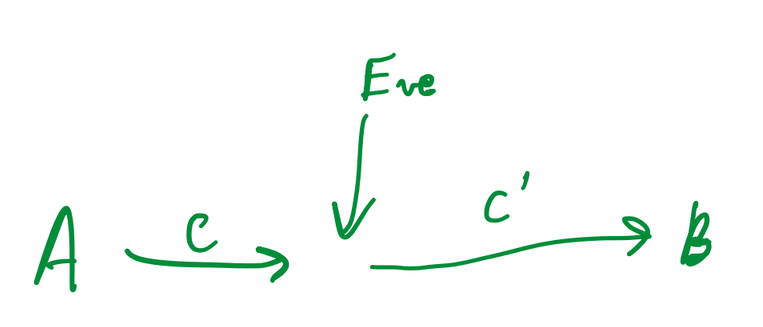
Tuesday, 11 October 2022 10:39

block ciphers + modes

counter mode turns block cipher into stream cipher

stream ciphers (from video)

give $Enc(k, m) = c$



in stream cipher,

Eve can modify c (still doesn't know from what & to what), she can also make up messages, e.g. send random c' or resend old c.

We need protection of integrity and authenticity

Message authentication codes (MACs)

MACs achieve both. The following is due to Wegman & Carter. There are many other MACs, e.g. HMAC

A & B share key k, derive r_1, r_2, r_3, r_4, r_5 which will be used for all ciphertexts & $s_1, s_2, s_3, \dots, s_{100}$ to be used for $c_1, c_2, c_3, \dots, c_{100}$.

Do not re-use any s_i !

The authentication tag on c_i is $t_i = (r_1 c_{i1} + r_2 c_{i2} + r_3 c_{i3} + r_4 c_{i4} + r_5 c_{i5}) \bmod p + s_i \bmod n$ assuming c_i has 5 blocks

for prime p and $n < p$, e.g. $p = 1000003$, $n = 1000000$

both p and n are publicly known.

A sends (c_i, t_i).

B recomputes t_i and accepts if equal to the value sent to him by Alice

A guessing Eve has a chance of $\leq \frac{2}{1000000}$ by picking random t'. (at most / less or equal to; because in some cases, we get the same number mod n, as $n < p$)

Somewhat more efficient: pick r,

put $r_i = r^{6-i}$

$t_i = (r^5 c_{i1} + r^4 c_{i2} + \dots + r c_{i5}) \bmod p + s_i \bmod n$ (exponent 1, not 0!)

Compute this via Horner's scheme:

$$((((r c_{i1} + c_{i2}) r + c_{i3}) r + c_{i4}) r + c_{i5}) r$$

This means we can also send longer messages.

s_i still needs to be unique; fixed-length tag

Are there More attacks which became possible?

Eve sees several (c_i, t_i) pairs, should come up with new (c', t')

Take some $t' = t_1$

aim for same s_1, need to have $\sum_{j=1}^{5} c_{1j} r^{6-j} = \sum_{j=1}^{5} c'_j r^{6-j}$

then (t_1, c') is valid

We are lucky if $\sum (c'_j - c_{1j}) r^{6-j} \equiv 0 \bmod p$

has at most 5 roots mod p (note: our field is not complex)

Eve can pick c' such that this has 5 roots, $\frac{5}{1000000}$ chance. This implicitly tries 5 values for r.

Better: look at $\left(\sum_{j=1}^{5} (c'_j - c_{1j}) r^{6-j}\right) \cdot \left(\sum_{j=1}^{5} (c'_j - c_{1j}) r^{6-j} + 1000000\right) \cdot \left(\sum_{j=1}^{5} (c'_j - c_{1j}) r^{6-j} - 1000000\right)$

This has degree 15 $\Longrightarrow$ so 15 choices of r tried at once if c' is picked so that this factors.

We can also change t_i to some $t' \neq t_i$; this just adds $t' - t_1$ to each of the three terms $\rightarrow$ same degree

5 comes from having 5 blocks, 3 comes from n-p

Note: we want s_i & $r < n$

Poly 1305 is a MAC of this type with $p = 2^{130} - 5$

$n = 2^{128}$

Some small technicalities: s_i = AES-CTR $(k, nonce_i)$ (AES in counter mode)

a nonce is a number used only once

$nonce_i$ is randomly pick

Send $(c, t, nonce)$

Always include MAC!!! so you can verify who sent the message

Use authenticated encryption

---

public key crypto → RSA encryption & signatures

security notion of PKE (public-key encryption)

- OW: one-wayness → can't get m from c
- IND: indistinguishability → attacker gets to pick m_0 & m_1, receives encryption of one of them, cannot guess correctly with more than 50% chance.

attacker powers

- KOA: key only attack
- CPA: chosen plaintext attack → i.e. attacker can see encryptions of messages of their chosen
  - → minimum power when attacking PKE
- CCA: chosen ciphertext attack → attacker can ask for decryption of ciphertexts of their choice
- CCA-I: decryptions in first phase only
- CCA-II: decryptions always

Key-Gen: pick primes $p \neq q$ of about the same size.

compute $n = p \cdot q$

pick e with $\gcd(e, (p-1)\cdot(q-1)) = 1$

typically $e = 2^{16} + 1$

compute $d \equiv e^{-1} \bmod \varphi(n)$ (compute with XGCD (extended Euclidean algorithm))

for $\varphi(n) = (p-1)(q-1)$ (Euler phi function / Euler totient function)

public key (n, e) private key (n, d)

Enc: $c \equiv m^e \bmod n$

Dec: $m \equiv c^d \bmod n$

This works by Fermat's little theorem.

Does RSA (schoolbook version) offer IND-CPA? No, Enc is deterministic, just encrypt m_0 and check for a match.

How about OW-CCA? (-II) Given c, can ask for decryptions of $c' \neq c$, output plaintext.

take $c' = 2c$, decryption is $(2c)^d \bmod n \equiv 2^d \cdot m$

take $c' = 2^e c$ (encryption of 2), decryption is $(2^e c)^d \bmod n \equiv 2^{de} \cdot m \equiv 2m \bmod n$

We can ask our CCA-oracle for decryption of $c' \equiv 2^e \cdot c \bmod n$, get back $m' = 2m \bmod n$, divide by 2 to get $m \equiv \frac{m'}{2} \bmod n$

This uses that schoolbook RSA is homomorphic, i.e. $Enc(m_1 \cdot m_2) = Enc(m_1) \cdot Enc(m_2)$. → This breaks OW-CCA.

RSA-OAEP: optimal asymmetric encryption padding

In practice, we are going to use RSA with padding, e.g. RSA-OAEP