CS558 Network Security

Lecture 2: Review Day

hello

BOSTON UNIVERSITY

Pre-req’s Reminder

Step 1: Reviewing Networks

- Interfaces between protocols
- Implicit trust assumptions
- General note: abstractions and complexity

Step 2: Reviewing Cryptography

- Notation!

$$\boxed{Pr}[Enc_k(m) = c] = Pr[Enc_K(m') = c]$$

$\{0,1\}^\lambda$

$m = m' || 0$

Pr [Enck(m) = c] = Pr [Enck(m’) = c]

$M = \{m = m'||0 \mid m' \in \{0, 1\}^{\ell-1}\}$, $\quad K = \{0, 1\}^{\ell}$

Sample at Random

$K \xleftarrow{\$} \{0,1\}^{\ell}$

BOSTON UNIVERSITY

Secret Key vs Public Key

receiver Sender Share a key $k \xleftarrow{\$} \{0,1\}^\lambda$

$c \leftarrow Enc_k(m)$ $Enc(k, m)$

$m \leftarrow Dec_k(c)$ $m \leftarrow Dec(k, c)$

Encryption: Secret Key vs Public Key

$pk, sk \leftarrow KeyGen(1^\lambda; r)$

$c \leftarrow Enc_{pk}(m)$

$m \leftarrow Dec_{sk}(c)$

Diffie-Hellman Key Exchange

$g \in \cancel{G}\ \mathbb{Z}_p$

Sender

$a \xleftarrow{\$} \mathbb{Z}_p$

$\xrightarrow{\quad g^a \quad}$

Receiver

$b \xleftarrow{\$} \mathbb{Z}_p$

$(g^b)^a = g^{ab}$

$\xleftarrow{\quad g^b \quad}$

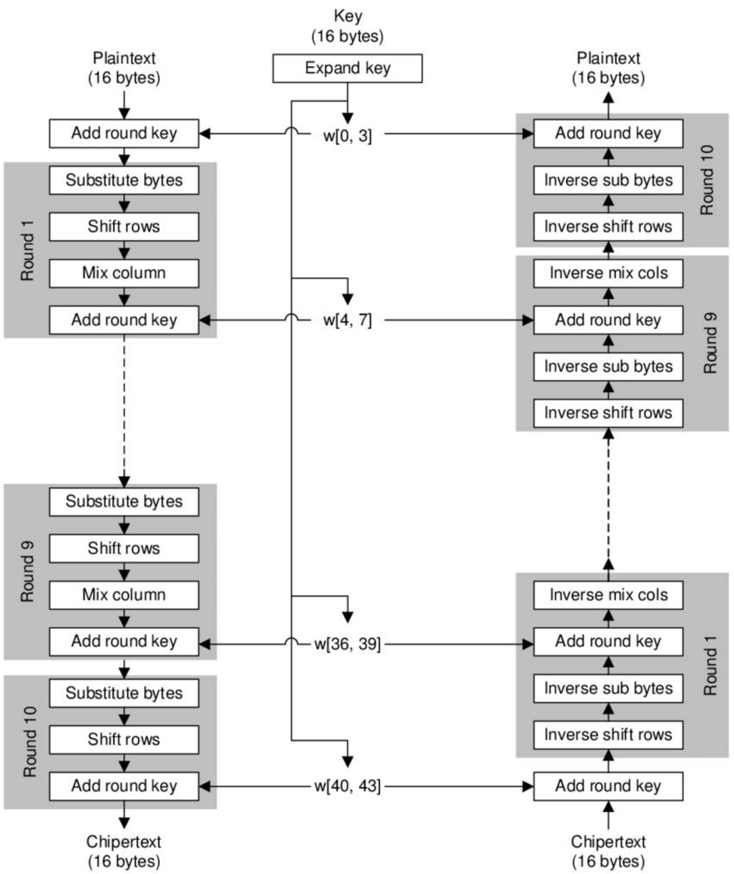
$(g^a)^b = g^{ab}$

g^{a+b}

BOSTON UNIVERSITY

AES

El Gamal (Half a KE)



Speeed

```
OpenSSL 1.1.1i  8 Dec 2020
built on: Wed Jan 13 03:19:58 2021 UTC
options:bn(64,64) rc4(16x,int) des(int) aes(partial) idea(int) blowfish(ptr)
compiler: clang -fPIC -arch x86_64 -O3 -Wall -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_CPUID_OBJ -DOPENSSL_IA32_SSE2
-DOPENSSL_BN_ASM_MONT -DOPENSSL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DKECC
AK1600_ASM -DRC4_ASM -DMD5_ASM -DAESNI_ASM -DVPAES_ASM -DGHASH_ASM -DECP_NISTZ256_ASM -DX25519_ASM -DPOLY1305_
ASM -D_REENTRANT -DNDEBUG
The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes  16384 bytes
aes-128 cbc     203039.25k   203253.16k   212326.06k   212439.04k   211148.12k   205967.15k
aes-256 cbc     140512.52k   147430.40k   144983.23k   146337.87k   151508.02k   158121.98k
                  sign    verify    sign/s verify/s
rsa 2048 bits 0.000598s 0.000028s   1673.2  35308.3
                  sign    verify    sign/s verify/s
dsa 2048 bits 0.000393s 0.000347s   2546.3   2883.7
```

~200x Faster

BOSTON UNIVERSITY

Authenticity: Secret Key vs Public Key

MAC — Message Authentication Codes → digital signatures

$t \leftarrow MAC_k(m)$

$\{0,1\} \leftarrow Verify_k(m, t)$

$t' \leftarrow MAC_k(m)$

$t \stackrel{?}{=} t'$

$\sigma \leftarrow Sign_{sk}(m)$

$\{0,1\} \leftarrow Verify_{pk}(m, \sigma)$

WHAT you sign matters

Exercise 3. An airline uses *manifests* to determine which passenger should be on which flight.

The airline has the secret key k. Each manifest consists of:

- The flight number f and its date and time d
- A MAC $t = \mathsf{MAC}_K(f||d)$.
- The name of the 1^{st} passenger p_1, and a digital signature $t_1 = \mathsf{MAC}_{SK}(p_1)$.
- The name of the 2^{nd} passenger p_2, and a digital signature $t_2 = \mathsf{MAC}_{SK}(p_2)$.

  $\vdots$
- The name of the n^{th} passenger p_n, and a digital signature $t_n = \mathsf{MAC}_{SK}(p_n)$.

Imagine that this is just a ticket you get + present at the gate

not super important that this is a different key

Notice that n will be different for each flight.

The manifest is checked, using the key k, as passengers board the flight.

1. **(4 points).**
   Suppose you can intercept and modify manifests before they arrive at each flight.
   Explain how you can travel to Tokyo for the cost of a flight to Chicago.

BOSTON UNIVERSITY