Dec3

Thursday, December 3, 2020 1:46 PM

Recall

$\text{HASH}: \{\text{bitstrings}\} \longrightarrow \{0, 1\}^k$

1) Easy to compute.

2) Hard to invert

↳ Hard to identify which bitstrings have a given Hash.

Morally speaking (**)

Changing one bit in input should have a 50% of changing a given bit in output.

Applying Hash functions

① Hashing documents to sign.

② Random number generators (RNG)

(pseudo random number generators). (PRNG).

Every system used these.

- Elgamal (& MV-elgamal).
- DSA (choose random signing exponent).
- RSA (finding random large prime).
- Miller-Rabin (probably prime).

A perfect RNG

Consider a radioactive^{R} substance w/ a half-life of H seconds.

↳ If I have M atoms. After H seconds, M/2 atoms remain.

↪ 1 atom has exactly a 50% chance of decaying in H seconds.

Algorithm to get a random bitstring

1) Focus on a particular atom of R.
2) Wait H seconds.
3) If atom has decayed, next bit $= 1$
   else $= 0$
4) repeat.

Certainly not very applicable in practice.

In practice: Pseudo Random Number Gen

A function that produce $s_0, s_1, s_2, \ldots$ which appear random.

Example of possible framework

$f(X,Y)$ some function.

$S \leftarrow$ a "truly" random seed (secret)

(as random as possible mixing of least sig. digits of various system parameters).

$R_0 = f(0,S),\ R_1 = f(1,S),\ R_2 = f(2,S), \ldots$

$R = R_0 \| R_1 \| R_2 \| \ldots$

Impt properties (Secure PRNG).

① If Eve knows first k bits of Alice's bitstring, she should have no more than a 50% of predicting (k+1)st bit.

② If Eve knows $R_t, R_{t+1}, \ldots$ this doesn't help find $R_0, R_1, \ldots, R_{t-1}$.

Example

Build a PRNG out of a hash.

S = random seed. (works b/c of moral **)

$$R_i = HASH(i \| S)$$

Even if Eve knows HASH, she cannot recover S (hard to invert).
w/o S no way of finding $R_{i+1}, \ldots$)

Hashes as protection from the chosen ciphertext attack.

Recall:

Given $m \xrightarrow{e} c$ & $c \xrightarrow{d} m$

Chosen Cipher Attack

Eve has access to a decryption oracle an can therefore apply d to a cipher of her choice.

---

3 ways to exploit this

$\boxed{1}$ Bob encrypts m. Eve intercepts $c = e(m)$.
Eve tells the oracle to decrypt c.
Eve recovers $m = d(c)$.

Obvs

Security Protocol:

Computer won't decrypt cipher's it has already decrypted (w/ some extra checks).

$\boxed{2}$ Bob encrypts m. Eve intercepts $c = e(m)$.

Eve masks c. She choose a mask $\underline{k}$.

Computs $c' = e(k) \cdot c$. Oracle decrypt c'.

$$d(c') = d(e(k) \cdot c) = d(e(k)) \cdot d(c)$$
$$= k \cdot m$$

Eve compte $k^{-1}(km) = m$.

Examples

a) **RSA** $\quad \overset{\text{pub}}{N} = pq \quad \overset{\text{pub}}{e} \quad \overset{\text{priv.}}{d} = e^{-1} \bmod (p-1)(q-1)$

Bob encrypts $m \rightsquigarrow c = m^e \bmod N$.

Eve intercepts c.

Eve masks c, choose k, computes

$$c' = k^e \cdot c \bmod N.$$

Oracle decrypts c'

$$(c')^d = (k^e \cdot c)^d = k^{ed} c^d$$
$$= k\,m \quad \bmod N$$

Eve recover $m = k^{-1}(km)$.

(b) ElgaMal

$\underset{\text{pub}}{p} \qquad \underset{\text{pub}}{g \in \mathbb{F}_p^*} \qquad \underset{\text{priv.}}{a} \qquad \underset{\text{pub}}{A} = g^A \bmod p$

Bob Encrypt $m \in$

Random $\underset{\uparrow}{k}$. Compute $c_1 = g^k \bmod p$

$c_2 = A^k m \bmod p$

(c_1, c_2)

$$\left(\text{Alice decrypts} \quad ((c_1)^a)^{-1} c_2 = (g^{ka})^{-1} g^{ak} \cdot m \equiv m \right)$$

Eve Intercepts (c_1, c_2)

$$\left.\begin{array}{l}\text{Chooses random } k' \\ \text{Chooses random } m'\end{array}\right\}\text{mask}$$

$$c_1' \equiv c_1 \cdot \underline{g}^{k'} \qquad c_2' \equiv c_2 \cdot \underline{A^{k'} \cdot m'} \qquad \text{mod } p$$

Oracle decrypts

$$((c_1')^a)^{-1} \cdot c_2' = ((\underline{g}^k \underline{g}^{k'})^{\underline{a}})^{-1} \cancel{g^{ak}} m \, \cancel{g^{ak'}} m'$$

$$= mm'$$

Next Eve recovers $m = (mm')m'^{-1} \text{ mod } p$

Remark / Exercise

MV - Elgamal is also vulnerable to this attack in similar way.

3 Eve decrypts a chosen ciphertext that reveals to her the entire decryption protocol. (Makes all future messages insecure).

Examples

- Symmetric Ciphers

  $$d_k(1) = k^{-1} \text{ mod } p$$

  ↑ multiply by this is decrypt

- Encryption based on taking square roots.

$$\begin{tikzcd} \text{Factoring } pq \arrow[r] & \text{Take square roots mod } pq. \arrow[l, dashed, bend left, "\text{choose right value.}"] \end{tikzcd}$$

Suppose decryption involves taking a square root.

Eve decryption oracle ⇝ Compute square roots

⇝ Factor pq

⇝ Compute any square root.

Broken algorithm.

Provable Security (A quick tangent).

Hard math problem ⇝ Trap door function ⇝ Assymetric Cryptosystem.

We've proved

Solving hard problem ⟹ Breaking cryptosystem.

① Discrete Log over $\mathbb{F}_p^*$ ⟹ Diffie Hellman, Elgamal, DSA

② Factoring $N = pq$ ⟹ RSA

⇓

③ Square roots mod pq ⟹ Goldwasser-Micali, Rabin Sqrt

④ Discrete log over E $\Longrightarrow$ ECDHP
ECElgamal
MV-Elgamal
ECDSA . . .

For 1,2,4 no proof in other direction.

e.g./ unknown that a solution to RSA leads to a solution to int. fact problem.

Proving Equivalence means studying security of system = studying hard math problem.

Goal

Find the precise hard math problem @ core of your cryptosystem.

Rabin sqrt $\Longleftrightarrow$ sqrts mod pq.

Paradox

Having this proof of security leaves you vulnerable to a chosen plaintext attack.

(Assuming there's an alg to solve hard problem from decryption).

Sketch

$$\begin{tikzcd}
\text{Hard Problem} \arrow[r, shift left, "alg^*"] & \text{Crypto} \arrow[l, shift left, "alg^*"]
\end{tikzcd}$$

Hard Problem (Chosen params $\mathcal{P}$) → Eve uses Chosen cipher to decrypt d(c)

d(c) → * Eve solves hard problem

Eve solves hard problem —*→ Eve has decryption function

Good motivation to have extra protection against chosen ciphertext attack.

$$\mathcal{M} \xrightarrow{e} \mathcal{C}$$

$$\mathcal{C} \xrightarrow{d} \mathcal{M}$$

Eve has an oracle who can decrypt a chosen ciphertext $c \in \mathcal{C}$.

Idea

* Decryption only work if c comes from a valid or meaningful ciphertext.
(Eve doesn't have a lot of control over what c will decrypt to)

* Hard for eve to detect whether a given c has meaninful decryption.

→ If eve chooses m then $c = e(m)$ will be meaninful... but there's no new info from decrypting this.
Just chosen plaintext attack which PKC are strong against.

* Define what meaninful means while making it hard to detect on

Ciphers

Hashing & Message Padding

Lemma: $\oplus \leftarrow$ xor. α, α' bitstrings

$$(\alpha \oplus \alpha') \oplus \alpha' = \alpha.$$

Pf/ Reduce to case β, β' are bits

$$\beta \oplus \beta' \oplus \beta' \equiv \beta + \beta' + \beta' \mod 2$$
$$\equiv \beta + 2\beta' \mod 2$$
$$\equiv \beta \mod 2.$$

Optimal Asymmetric Encryption Padding

Have $e : \mathcal{M} \longrightarrow \mathcal{C}$

$d : \mathcal{C} \longrightarrow \mathcal{M}$

Alice & Bob Fix

2 hash functions $\mathcal{H}$ and $\mathcal{G}$.

An integer k.

Alice

Bob

Message m

Random bitstring r.

$$b = \mathcal{G}(r) \oplus (m \| \underbrace{00 \cdots 0}_{k})$$

$$c = e(b \| \mathcal{H}(b) \oplus r)$$

send

Decrypt

$$d(c) = b \| \underbrace{\mathcal{H}(b) \oplus r}$$

peel of b. $\leftarrow$

Compute $\mathcal{H}(b)$

Comput

$$\mathcal{H}(b) \oplus (\mathcal{H}(b) \oplus r)$$
$$= r$$

Computes $\mathcal{G}(r)$. $\leftarrow$

$$\mathcal{G}(r) \oplus b$$
$$= \mathcal{G}(r) \oplus \mathcal{G}(r) \oplus (m \| 0 \cdots 0)$$

$$= m \,\|\, \overbrace{\underline{0000 \cdots 0}} = m'$$

Ends w/ k-zeros so acceptable plaintext!

Notice

Use of Hash functions means change one bit of r or m', each bit of c has a 50% of changing... & vice versa!!

If Eve changes 1 bit of c, m' has $1/2^k$ chance of finishing w/ k zeros.

Decryption will only return messages w/ k 0's @ end.

Random ciphertext have very low probability of this happening.

Moral

Hard for eve to know which ciphertexts end in k 0's, b/c of difficulty of inverting Hash functions,

This thwarts chosen ciphertext attack (*K + obvious protocol).

Rmk

OAEP was in use almost a decade before it was "solve" / "broken"

THANKS :)

- *Good luck on TH2 / Final
- * Fill out course evals!!
- *office hours continue through finals week.