

# MI-KRY – Advanced Cryptology

Meet-in-the-middle

Ing. Jiří Buček



České vysoké učení technické v Praze  
Fakulta informačních technologií  
Katedra informační bezpečnosti

©2018 Jiří Buček.

[bucekj@fit.cvut.cz](mailto:bucekj@fit.cvut.cz)

# Lecture outline

- Brute force vs. meet-in-the-middle attack
- Meet-in-the-middle attack on 3DES
- Meet-in-the-middle attack on RSA without padding

# Brute force attack

Function  $f : \{1, 2, \dots, N\} \mapsto \{1, 2, \dots, N\}$  is fixed. Attacker knows  $y$ , where  $y = f(x)$ , and needs to find  $x$ .

- Find a preimage of a hash function  $f$
- Find the key for a fixed plaintext  $P$ , ciphertext  $C$ :

$$f(k) = C = E_k(P)$$

- Find the secret message  $M$  knowing the ciphertext  $C$  and public key  $PK$ :

$$f(M) = C = E_{PK}(M)$$

Assume it is difficult to find  $f^{-1}$ , so that  $f^{-1}(y) = x$ .

Brute force attack: Search exhaustively for  $x$  until  $y = f(x)$ . Complexity  $O(N) = O(2^n)$ , memory negligible.

# Brute force attack on double encryption

Let us have an encryption transformation  $c = E_K(p)$  that is composed of two subsequent transformations  $E'_{K_1}$  and  $E''_{K_2}$ , so that  $c = E_K(p) = E''_{K_2}(E'_{K_1}(p))$ .

The subkeys are defined as  $K_1 \in \{0, 1\}^n$ ,  $K_2 \in \{0, 1\}^m$ .

Decryption:  $p = D_K(c) = D'_{K_1}(D''_{K_2}(c))$

Goal: Recover  $K_1$  and  $K_2$ , knowing  $p$  and  $c$  (known plaintext attack).

Brute force attack: Exhaustive search for  $K_1$  and  $K_2$ , time  $O(2^{n+m})$  encryptions.

# Meet-in-the-middle attack

We can exploit the double-encryption composition to find a **collision** in the middle of the computation:

$$\begin{aligned}c &= E''_{K_2}(E'_{K_1}(p)) \\ D''_{K_2}(c) &= D''_{K_2}(E''_{K_2}(E'_{K_1}(p))) \\ D''_{K_2}(c) &= E'_{K_1}(p)\end{aligned}$$

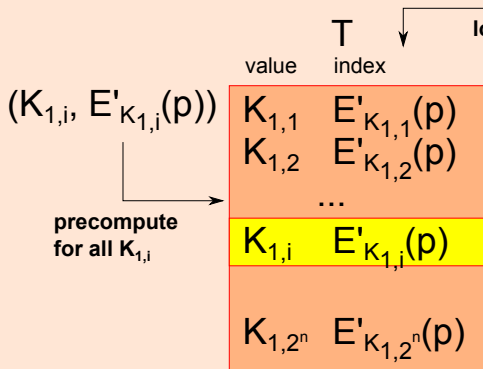
## Meet-in-the-middle

- First, precompute all possible  $E'_{K_1}(p)$  for the plaintext and store the pairs  $(K_1, E'_{K_1}(p))$  in a lookup table  $T$ .
- Then, find  $K_2$  such that  $D''_{K_2}(c)$  is in  $T$ , i.e.  $\exists K_1, (K_1, D''_{K_2}(c)) \in T$ .
- Found  $(K_1, K_2)$  is the key of  $c = E''_{K_2}(E'_{K_1}(p))$ .
- Verify  $(K_1, K_2)$  on other ciphertexts, continue searching if incorrect.

The precomputation phase requires  $O(2^n)$  encryptions and  $O(2^n)$  memory. The lookup phase requires  $O(2^m)$  encryptions and lookups. The resulting time complexity is  $O(2^n + 2^m)$ , as opposed to  $O(2^{n+m})$  for the simple exhaustive search.

# Meet-in-the-middle attack

## Precomputation phase



time:  $O(2^n)$

memory:  $O(2^n)$

## Lookup phase

$$D''_{K_{2,j}}(c)$$

time:  $O(2^m)$

found!  $E'_{K_{1,i}}(p) = D''_{K_{2,j}}(c)$

verify

on more pairs

$$E''_{K_{2,j}}(E'_{K_{1,i}}(p)) = c$$

$$(K_1, K_2) = (K_{1,i}, K_{2,j})$$

# Meet-in-the-middle attack example: 3DES-EDE

Let  $E_K$  and  $D_K$  be DES encryption and decryption, respectively, with  $K \in \{0, 1\}^{56}$ . Consider the usual EDE variant of  $3DES_{K_{123}}$ , where  $K_{123} = (K_1|K_2|K_3)$ :  $c = 3DES_{K_{123}}(p) = E_{K_3}(D_{K_2}(E_{K_1}(p)))$ .

$$\begin{aligned}c &= E_{K_3}(D_{K_2}(E_{K_1}(p))) \\ D_{K_3}(c) &= D_{K_2}(E_{K_1}(p)) \\ E_{K_2}(D_{K_3}(c)) &= E_{K_1}(p)\end{aligned}$$

Meet-in-the-middle: First, precompute all possible  $E_{K_1}(p)$  for the plaintext and store them in a lookup table  $T$ . Then, find  $(K_2, K_3)$  such that  $\exists K_1 : (K_1, E_{K_2}(D_{K_3}(c))) \in T$ . Then,  $(K_1, K_2, K_3)$  is likely the correct key of  $c = E_{K_3}(D_{K_2}(E_{K_1}(p)))$ .

The precomputation phase requires  $O(2^{56})$  encryptions and  $O(2^{56})$  memory. The lookup phase requires  $O(2^{112})$  encryptions and lookups. The resulting time complexity approx.  $O(2^{112})$ , as opposed to  $O(2^{168})$  for the simple exhaustive search. Therefore we must consider 3DES to have 112-bit security even though  $K_1 \neq K_2 \neq K_3$ .

# Meet-in-the-middle: RSA without padding

Consider a randomly chosen 64-bit message  $M$  (for example, a session key) that is encrypted using RSA with a public key  $(n, e)$ . Assume  $e$  is large enough to hinder the attack using integer  $e$ -th root.

$$C = |M^e|_n, M < 2^m$$

Goal: Recover the  $m$ -bit secret message  $M$ , knowing  $C, n, e$ .

With high probability, the random  $M$  is composite, so that  $M = M_1 M_2$ , where  $M_1 < 2^{m_1}$ ,  $M_2 < 2^{m_2}$ . For example, if  $m = 64$ , then  $M$  is composed of 32-bit numbers with 18% probability, and of at most 33-bit numbers with 29% probability.



# Meet-in-the-middle: RSA without padding 2

Due to the multiplicative homomorphism of RSA,

$C = |M^e|_n = |(M_1 M_2)^e|_n = |M_1^e M_2^e|_n$ , therefore  
(assuming  $\gcd(M_1, n) = 1$ )

$$|C M_2^{-e}|_n = |M_1^e|_n$$

We can search for a collision, looking for  $M_1$  and  $M_2$  separately. The attack is not guaranteed to always succeed, but it will succeed with high probability.

# Meet-in-the-middle: RSA without padding 3

Attack (meet-in-the-middle):

- Precompute all possible  $|M_1^e|_n$  and store the pairs  $(M_1, |M_1^e|_n)$  in a lookup table  $T$ .
- Then, find  $M_2$  such that  $|C M_2^{-e}|_n$  is in  $T$ , i.e.  
 $\exists M_1 : (M_1, |C M_2^{-e}|_n) \in T$ .
- Found  $(M_1, M_2)$  likely gives the secret  $M = M_1 M_2$ .

We do not need to store the whole  $|M_1^e|_n$  in  $T$ , it suffices to store enough bits to distinguish between different numbers.  $2^{\max(m_1, m_2)}$  least significant bits should be enough.

The attack requires  $2^{m_1+1} \max(m_1, m_2)$  bits of memory and takes  $2^{m_1} + 2^{m_2}$  modular exponentiations (including precomputation).

# Similar attacks

Meet-in-the-Middle is a generic space-time tradeoff attack. Other similar attacks include:

- Time-memory tradeoff attack on block, stream ciphers – Hellman
- Time-memory tradeoff attack on hash function preimages
  - ▶ Original attack by Martin Hellman
  - ▶ Rainbow tables attack by Philippe Oechslin
- Baby step – giant step algorithm for DLP

For similar time-memory tradeoff attacks on hash functions and stream ciphers, including rainbow tables, see the MI-BHW (Security and Hardware) course, Lecture 10.

# Bibliography



W. Diffie, M.E. Hellman, "Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard," Computer, vol. 10, no. 6, pp. 74-84, June, 1977



Boneh, Dan, Antoine Joux, and Phong Q. Nguyen. "Why textbook ElGamal and RSA encryption are insecure." Advances in Cryptology—ASIACRYPT 2000. Springer Berlin Heidelberg, 2000. 30-43., extended abstract at <http://www.ssi.gouv.fr/archive/fr/sciences/fichiers/lcr/bojong00.pdf>



S. Moore: Meet-in-the-Middle Attacks, <http://stephanemoore.com/pdf/meetinthemiddle.pdf>



Paar, Christof, and Jan Pelzl. Understanding cryptography: a textbook for students and practitioners. Springer Science & Business Media, 2009.