Applied Cryptography and Computer Security

CSE 664 Spring 2020

Lecture 7: Advanced Encryption Standard (AES)

Department of Computer Science and Engineering
University at Buffalo

Lecture Outline

- Last time:
  - block ciphers
  - Data Encryption Standard
  - attacks on DES
  - double and triple DES
- This lecture:
  - Advanced Encryption Standard
  - cipher details

Advanced Encryption Standard (AES)

- In 1997 NIST made a formal call for an unclassified publicly disclosed encryption algorithm available worldwide and royalty-free
  - the goal was to replace DES with a new standard called AES
  - the algorithm must be a symmetric block cipher
  - the algorithm must support (at a minimum) 128-bit blocks and key sizes of 128, 192, and 256 bits
- The evaluation criteria were:
  - security
  - speed and memory requirements
  - algorithm and implementation characteristics

AES

- In 1998 15 candidate AES algorithms were announced
- They were narrowed to 5 in 1999: MARS, RC6, Rijndael, Serpent, and Twofish
  - all five were thought to be secure
- A more thorough evaluation was performed
- In 2000 NIST announced that Rijndael was selected as the AES
- In 2001 AES was published for public review and comments and adopted later that year (published in FIPS 197)
- The selection process for the AES was very open

AES

- Rijndael
  - invented by Belgian researchers Deamen and Rijmen
  - designed to be simple and efficient in both hardware and software on a wide range of platforms
  - supports different block sizes (128, 192, and 256 bits)
  - supports keys of different length (128, 192, and 256 bits)
  - uses a variable number of rounds
    - $Nr = 10$ if both keys and block sizes are 128
    - $Nr = 12$ if max of block and key sizes is 192
    - $Nr = 14$ if max of block and key sizes is 256

AES

- During encryption:
  - the block is copied into the state matrix
  - the state is modified at each round of encryption and decryption
  - the final state is copied to the ciphertext

in_0	in_4	in_8	in_{12}
in_1	in_5	in_9	in_{13}
in_2	in_6	in_{10}	in_{14}
in_3	in_7	in_{11}	in_{15}

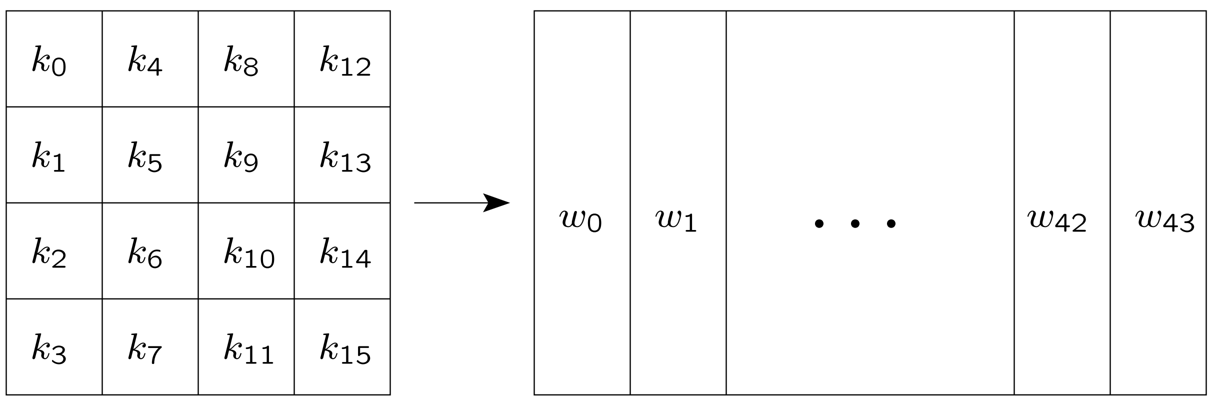
$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$

out_0	out_4	out_8	out_{12}
out_1	out_5	out_9	out_{13}
out_2	out_6	out_{10}	out_{14}
out_3	out_7	out_{11}	out_{15}

AES

- The key schedule in AES
  - the key is treated as a 4×4 matrix as well
  - the key is then expanded into an array of words
  - each word is 4 bytes and there are 44 words (for 128-bit key)
  - four distinct words serve as a round key for each round



AES

- Rijndael doesn't have a Feistel structure
  - 2 out of 5 AES candidates (including Rijndael) don't use Feistel structure
  - they process the entire block in parallel during each round
- The operations are (3 substitution and 1 permutation operations):
  - SUBBYTES: byte-by-byte substitution using an S-box
  - SHIFTROWS: a simple permutation
  - MIXCOLUMNS: a substitution using mod 2^8 arithmetics
  - ADDROUNDKEY: a simple XOR of the current state with a portion of the expanded key

AES

- At a high-level, encryption proceeds as follows:
  - set initial state $s_0 = m$
  - perform operation AddRoundKey (XORs k_i and s_i)
  - for each of the first $Nr - 1$ rounds:
    - perform a substitution operation SubBytes on s_i and an S-box
    - perform a permutation ShiftRows on s_i
    - perform an operation MixColumns on s_i
    - perform AddRoundKey
  - the last round is the same except no MixColumns is used
  - set the ciphertext $c = s_{Nr}$

AES

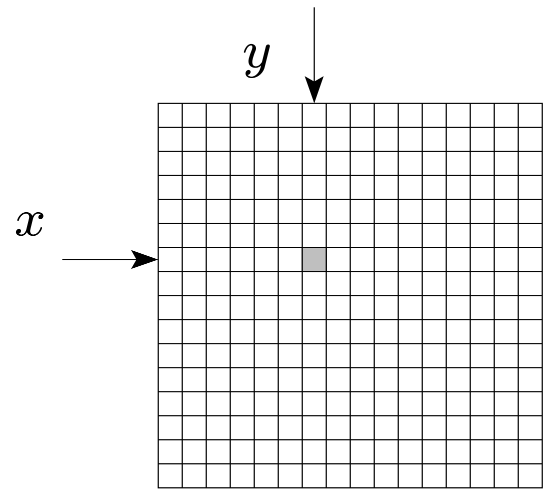
- More about Rijndael design. . .
  - AddRoundKey is the only operation that uses key
    - that's why it is applied at the beginning and at the end
  - all operations are reversible
  - the decryption algorithm uses the expanded key in the reverse order
  - the decryption algorithm, however, is not identical to the encryption algorithm

AES

- The SUBBYTES operation
  - maps a state byte $s_{i,j}$ to a new byte $s'_{i,j}$ using S-box
  - the S-box is a 16×16 matrix with a byte in each position
    - the S-box contains a permutation of all possible 256 8-bit values
    - the values are computed using a formula
    - it was designed to resist known cryptanalytic attacks (i.e., to have low correlation between input bits and output bits)

AES

- The SubBytes operation
  - to compute the new $s'_{i,j}$:
    - set x to the 4 leftmost bits of $s_{i,j}$ and y to its 4 rightmost bits
    - use x as the row and y as the column to locate a cell in the S-box
    - use that cell value as $s'_{i,j}$



  - the same procedure is performed on each byte of the state

AES

- The SHIFTROWS operation
  - performs circular left shift on state rows
    - 2nd row is shifted by 1 byte
    - 3rd row is shifted by 2 bytes
    - 4th row is shifted by 3 bytes

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$

$\longrightarrow$

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,1}$	$s_{1,2}$	$s_{1,3}$	$s_{1,0}$
$s_{2,2}$	$s_{2,3}$	$s_{2,0}$	$s_{2,1}$
$s_{3,3}$	$s_{3,0}$	$s_{3,1}$	$s_{3,2}$

  - important because other operations operate on a single cell

AES

- The MixColumns operation
  - multiplies the state by a fixed matrix

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

  - was designed to ensure good mixing among the bytes of each column
  - the coefficients 01, 02, and 03 are for implementation purposes (multiplication involves at most a shift and an XOR)

AES

- Decryption:
  - inverse S-box is used in SubBytes
  - inverse shifts are performed in ShiftRows
  - inverse multiplication matrix is used in MixColumns
- Key expansion:
  - was designed to resist known attacks and be efficient
  - knowledge of a part of the key or round key doesn't enable calculation of other key bits
  - round-dependent values are used in key expansion

AES

- Summary of Rijndael design
  - simple design but resistant to known attacks
  - very efficient on a variety of platforms including 8-bit and 64-bit platforms
  - highly parallelizable
  - had the highest throughput in hardware among all AES candidates
  - well suited for restricted-space environments (very low RAM and ROM requirements)
  - optimized for encryption (decryption is slower)

Encryption Modes

- Recall that encryption modes specify how messages longer than one block are encrypted and decrypted
- 4 modes of operation were standardized in FIPS Pub. 81 for DES
  - electronic codebook mode (ECB), cipher feedback mode (CFB), cipher block chaining mode (CBC), and output feedback mode (OFB)
- 5 modes have been approved by NIST for AES and other ciphers in 2001
  - the 4 above and counter mode

Bootstrapping Symmetric Encryption

- You can communicate a secret key to your friend by:
  - phone, (slow) mail, inviting her for dinner, ...
- We are going to use public key encryption to communicate the symmetric encryption key
- To agree on a secret symmetric key, the idea is:
  - pick a fresh secret key s and encrypt it with the friend's publicly known key pk as $\mathsf{Enc}_{pk}(s)$
  - the friend will be able to decrypt and use s, but nobody else