Block Ciphers

Modes of Use, DES and AES

Joshua E. Hill

Department of Mathematics, University of California, Irvine

Math 235A
October 29 & 31, 2012
November 14 & 16, 2012
`http://bit.ly/UOtalq`
v1.1

Talk Outline

Pleased to meet you...

Who am I, anyway?

- Ph.D. candidate.
  - Area: Algorithmic Algebraic Number Theory
  - Advisor: Daqing Wan
- B.S. Computer Science (Cal Poly, San Luis Obispo)
- Worked at an information security consulting firm for 10 years.
  - I evaluated security products against various government standards.
  - I've looked at many, many products.

The Story Thus Far

A block cipher is a family of bijective functions, indexed by the key $k \in \mathcal{K}$. We call these families "encrypt" (e_k) and "decrypt" (d_k):

$$e_k : \mathcal{M} \rightarrow \mathcal{C}$$

$$d_k : \mathcal{C} \rightarrow \mathcal{M}$$

- For block ciphers, $\mathcal{M}$ and $\mathcal{C}$ are the same set.
- Every element of these sets are b-bit (binary) strings for some fixed b.
- This length, b, is called the block size.

A Short, Sharp, Shock

We will discuss:

- A variety of common and (sometimes) reasonable-to-use block cipher modes of use (this class and next).
- Some explicit examples of block cipher construction (starting November 14, 2012):
  - DES
  - AES

Block Cipher Modes Travel Guide

We'll discuss a few of the more common and useful modes of use:

- Block oriented Confidentiality Modes:
  - ECB
  - CBC
- Stream-cipher-like Confidentiality Modes
  - OFB
  - CTR
- Data Integrity Modes
  - CBC-MAC
  - CMAC
- Combined Confidentiality / Integrity Modes
  - CCM

Subsection 1

Block oriented Confidentiality Modes

ECB Mode

- The basic cipher, without modification.
- Split the plaintext into blocks, and encrypt each block independently.

ECB Notes

- No inter-dependency between blocks.
- Bit errors render the rest of the block uncontrollably corrupted.
- Encryption of identical plaintext blocks under the same key yield identical ciphertext blocks.
  - Exposes plaintext structural information.
  - Susceptible to attacker blockwise modification.

ECB: A Dungeon Dispatch

- I am a member of the MAA
- In Nomathistan, being a member of the MAA is punishable by death.
- When traveling, I thus must encrypt my pro-math propaganda:

Math

is

Great!!!

- I am a member of the MAA
- In Nomathistan, being a member of the MAA is punishable by death.
- When traveling, I thus must encrypt my pro-math propaganda:

Math

is

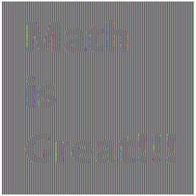
Great!!!

$\xrightarrow{\text{DES ECB}}$

- I am a member of the MAA
- In Nomathistan, being a member of the MAA is punishable by death.
- When traveling, I thus must encrypt my pro-math propaganda:

Math
is
Great!!!

DES ECB →



ECB: A Dungeon Dispatch

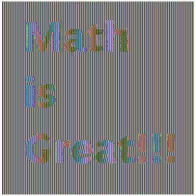
- I am a member of the MAA
- In Nomathistan, being a member of the MAA is punishable by death.
- When traveling, I thus must encrypt my pro-math propaganda:

Math

is

Great!!!

DES ECB →



- :-(

ECB: Post Mortem

What just happened?

- We mainly had two styles of blocks:
  - White parts

FF	FF	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	FF	FF	FF

ECB: Post Mortem

What just happened?

- We mainly had two styles of blocks:
  - White parts

FF	FF	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	FF	FF	FF

$\downarrow$

d2	13	19	a0	15	db	d6	71
d2	13	19	a0	15	db	d6	71
d2	13	19	a0	15	db	d6	71

ECB: Post Mortem

What just happened?

- We mainly had two styles of blocks:
  - Black parts

00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00

ECB: Post Mortem

What just happened?

- We mainly had two styles of blocks:
  - Black parts

00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00

↓

54	c0	c7	51	49	d9	12	4e
54	c0	c7	51	49	d9	12	4e
54	c0	c7	51	49	d9	12	4e

ECB: Post Mortem

What just happened?

- We mainly had two styles of blocks:
  - Black parts

00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00

↓

54	c0	c7	51	49	d9	12	4e
54	c0	c7	51	49	d9	12	4e
54	c0	c7	51	49	d9	12	4e

- Structural information is exposed to an adversary.

ECB: Message Order Matters!

- Blocks can be reordered.
- Reordering or repetition of blocks can change the message.
- Any ciphertext encrypted with the same key can be used as source material for the attacker.
- ECB security is dependent on the plaintext data formatting!

Plaintext
From: King Claud
ius To: The King
of England Plea
se help me kill
my nephew Hamlet
. Please send me
evidence with my
loyal chattel:
my men Rosencra
ntz & Guildenste
rn.

ECB: The Rosencrantz & Guildenstern Affair

AES ECB →

Plaintext	Ciphertext
From: King Claud	90d1dac87eca9f739b2fa23dff7af501
ius To: The King	66e6a94a67b88c471f82321e5d32f4e4
of England Plea	54e13d9dbfd2a391f23b3f7904e6f789
se help me kill	9c38b26e40c6a25000c145b49b783d42
my nephew Hamlet	ce62406ec7d8e2c21323083c4a2c2d62
. Please send me	ce95c814f1005e468f1f8a2eaa3ab52b
evidence with my	8a824c1b8ac2a007efc733ddc6684a3c
loyal chattel:	7aa0438c10f0d68114715094ba1e79c0
my men Rosencra	bd812a6b8b9b4e7f8abe36f067c9fb4c
ntz & Guildenste	3724d63b1f8555baa42347fbd2da793d
rn.	0b41dc57dd4b626372c244548e31871a

ECB: The Rosencrantz & Guildenstern Affair

Reordered Ciphertext
90d1dac87eca9f739b2fa23dff7af501
66e6a94a67b88c471f82321e5d32f4e4
54e13d9dbfd2a391f23b3f7904e6f789
9c38b26e40c6a25000c145b49b783d42
bd812a6b8b9b4e7f8abe36f067c9fb4c
3724d63b1f8555baa42347fbd2da793d
0b41dc57dd4b626372c244548e31871a
ce95c814f1005e468f1f8a2eaa3ab52b
8a824c1b8ac2a007efc733ddc6684a3c
7aa0438c10f0d68114715094ba1e79c0
ce62406ec7d8e2c21323083c4a2c2d62

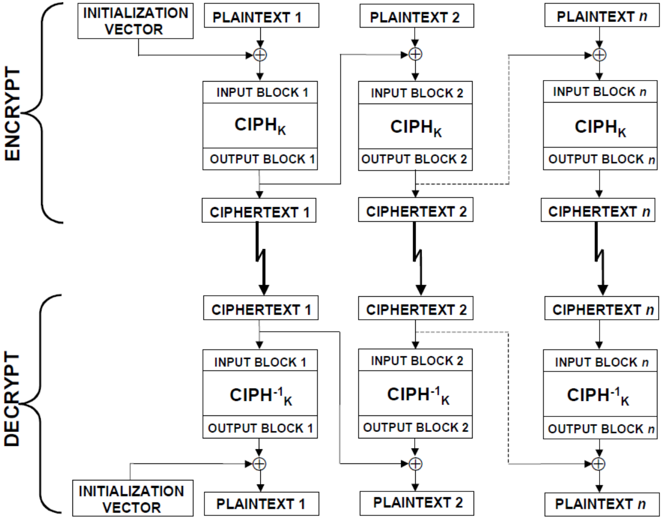
ECB: The Rosencrantz & Guildenstern Affair

Resulting Plaintext		Reordered Ciphertext
From: King Claud		90d1dac87eca9f739b2fa23dff7af501
ius To: The King		66e6a94a67b88c471f82321e5d32f4e4
of England Plea		54e13d9dbfd2a391f23b3f7904e6f789
se help me kill		9c38b26e40c6a25000c145b49b783d42
my men Rosencra	$\xleftarrow{(\text{AES ECB})^{-1}}$	bd812a6b8b9b4e7f8abe36f067c9fb4c
ntz & Guildenste		3724d63b1f8555baa42347fbd2da793d
rn.		0b41dc57dd4b626372c244548e31871a
. Please send me		ce95c814f1005e468f1f8a2eaa3ab52b
evidence with my		8a824c1b8ac2a007efc733ddc6684a3c
loyal chattel:		7aa0438c10f0d68114715094ba1e79c0
my nephew Hamlet		ce62406ec7d8e2c21323083c4a2c2d62

CBC Mode

- Uses a 1-block initial vector (IV)
- IV is XORed with plaintext before encryption
- For later blocks, uses the prior ciphertext as the IV

CBC Diagram



Source: NIST SP800-38A

CBC Notes

- Encrypting the same plaintext under the same key produces different ciphertext, so long as the block IV is different
  - Different IV choice
  - Different prior plaintext
  - In the instance of IV collision, structural information is revealed (as with ECB).
- IV must be unpredictable (but need not be secret) but IV integrity should be assured.
- No error should be issued if padding is invalid. (*n.b.* Vaundenay, 2002)
- In the event of an error:
  - Block associated with change is fully corrupted.
  - Next block has changes to plaintext that are the error XOR the original plaintext.
  - Future blocks uncorrupted.

CBC: The Rosencrantz & Guildenstern Affair

`IV = a73c7304715b7ab1a4ec61b72c495963`

Plaintext
`From: King Claud`
`ius To: The King`
` of England Plea`
`se help me kill `
`my nephew Hamlet`
`. Please send me`
`evidence with my`
` loyal chattel: `
` my men Rosencra`
`ntz & Guildenste`
`rn.`

- Want to change $M1$ =“`my nephew Hamlet`” to $M2$ =“`my butter cookie`”.
- Calculate $A = M1 \oplus M2 =$ `0000000c10041c0005002b0e02070c11`.
- XOR the prior ciphertext block with A.

CBC: The Rosencrantz & Guildenstern Affair

`IV = a73c7304715b7ab1a4ec61b72c495963`

Plaintext
`From: King Claud`
`ius To: The King`
` of England Plea`
`se help me kill `
`my nephew Hamlet`
`. Please send me`
`evidence with my`
` loyal chattel: `
` my men Rosencra`
`ntz & Guildenste`
`rn.`

AES CBC →

Ciphertext
`df93d586ecabd9d4cc22006514201d52`
`f7d51c26b7c715040d5efafcace90398`
`112d44070df51b8752eb34fec354b932`
`d0d1771177e57e6904516f1c0eea0381`
`965920f320efe797c18840fc15f9dd03`
`b08b245c3ea500291cb3b6e3a0e8e1ea`
`48fa04608cae01625ebe755be7f6fe47`
`3ca66358d5dae7f6c1eceeb9488d1749`
`77b9eebc040d659aaa169aaa92d2a141`
`48e807c85b57bedcbab671dbbf618a0d`
`0530a5411fc7b7757abcd962ccb20b09`

- Want to change $M1$ =“`my nephew Hamlet`” to $M2$ =“`my butter cookie`”.
- Calculate $A = M1 \oplus M2 =$ `0000000c10041c0005002b0e02070c11`.
- XOR the prior ciphertext block with A.

CBC: The Rosencrantz & Guildenstern Affair

Modified Ciphertext
df93d586ecabd9d4cc22006514201d52
f7d51c26b7c715040d5efafcace90398
112d44070df51b8752eb34fec354b932
d0d1771d67e16269015144120ced0f90
965920f320efe797c18840fc15f9dd03
b08b245c3ea500291cb3b6e3a0e8e1ea
48fa04608cae01625ebe755be7f6fe47
3ca66358d5dae7f6c1eceeb9488d1749
77b9eebc040d659aaa169aaa92d2a141
48e807c85b57bedcbab671dbbf618a0d
0530a5411fc7b7757abcd962ccb20b09

CBC: The Rosencrantz & Guildenstern Affair

Resulting Plaintext		Modified Ciphertext
From: King Claud		df93d586ecabd9d4cc22006514201d52
ius To: The King		f7d51c26b7c715040d5efafcace90398
of England Plea		112d44070df51b8752eb34fec354b932
..öÍ.æ6.õ.ÓRû.É{		d0d1771d67e16269015144120ced0f90
my butter cookie	$\xleftarrow{(\text{AES CBC})^{-1}}$	965920f320efe797c18840fc15f9dd03
. Please send me		b08b245c3ea500291cb3b6e3a0e8e1ea
evidence with my		48fa04608cae01625ebe755be7f6fe47
loyal chattel:		3ca66358d5dae7f6c1eceeb9488d1749
my men Rosencra		77b9eebc040d659aaa169aaa92d2a141
ntz & Guildenste		48e807c85b57bedcbab671dbbf618a0d
rn.		0530a5411fc7b7757abcd962ccb20b09

The Birthday Paradox, Writ Large

- We model most cryptographic primitives as random mappings.
- For many of our uses, it is interesting when a new output is equal to an old output; this is called a collision.
- What is the probability that some output is equal to a prior output after j outputs have been produced?
- Probability of "no collision" is easier to address, and then take the complement. After j outputs:

$$
\begin{aligned}
\Pr(\text{collision}) &= 1 - \prod_{k=0}^{j-1} \left(1 - \frac{k}{2^b}\right) \\
&= 1 - \frac{\left(2^b\right)\left(2^b - 1\right)\left(2^b - 2\right)\cdots\left(2^b - j + 1\right)}{2^{bj}} \\
&= 1 - \frac{\left(2^b\right)^{\underline{j}}}{2^{bj}}
\end{aligned}
$$

Let's ask Newton

- Recall $e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$; so long as x is close to 0, $e^x \approx 1 + x$.
- The terms in our product look like $1 - \frac{k}{2^b}$.
- To approximate we set $1 - \frac{k}{2^b} = 1 + x$ and find that $x = -\frac{k}{2^b}$.
- This yields the approximation:

$$\Pr(\text{collision}) \approx 1 - \prod_{k=0}^{j-1} e^{k/2^b} = 1 - \exp\left(-\frac{(j)(j-1)}{2^{b+1}}\right)$$

- So long as $j \ll 2^b$, this approximation remains quite reasonable.

How Much Wood...

Block Size	Pr(collision) 2^{-40}	2^{-20}	2^{-1}
64	2^{11}	2^{21}	2^{31}
128	2^{43}	2^{53}	2^{63}

Table : Allowed Outputs for Target Collision Probability

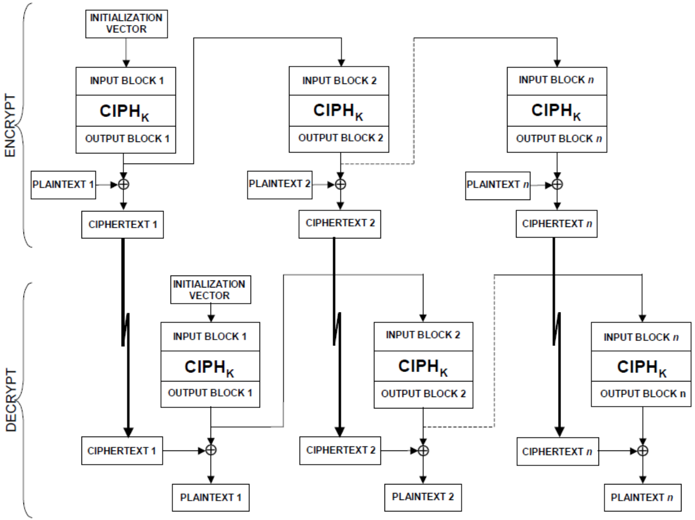
Subsection 2

Stream-cipher-like Confidentiality Modes

OFB Mode

- A stream cipher mode that can encrypt arbitrary blocks of data (though specified to work 1 block at a time)
- IV is the initial cipher input.
- Output of cipher is the keystream, and is XORed with the plaintext to create the ciphertext.
- Prior keystream becomes the next IV

OFB Diagram



Source: NIST SP800-38A

OFB Notes

- Uses cipher in encrypt mode only
- Keystream is in no way affected by the plaintext
- Anytime there is an IV collision under the same key, the produced keystream is the same.
- In the instance of (Key,IV) collision, *all future* plaintexts are revealed, and the corresponding past plaintexts.
- IV must be a nonce, that is, it must only occur once per key.
- In the event of an error:
  - Bitwise equivalent changes occur in the corresponding plaintext.
  - Blocks after the altered block are unaffected.
  - An attacker can perform bit-level targeted modification to the plaintext.

Cycles Within Cycles

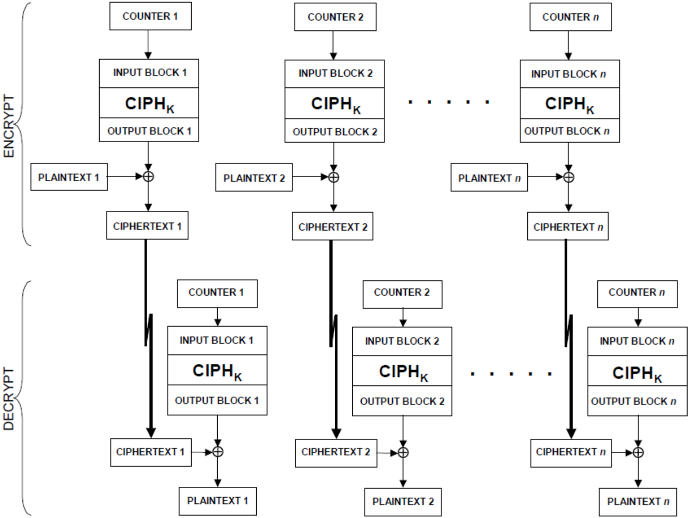
- There is absolutely no external input into the keystream generation process after initialization.
- Once a keystream block repeats, the keystream generation is trapped in a cycle, and cannot recover.
- If the attacker XORs two blocks of ciphertext that used the same keystream, the result is two plaintexts XORed (this is considered decrypted).
- Cycle detection isn't cheap or fast (and is thus not commonly done).
- How many blocks should be encrypted with a single key?

	Pr(collision)		
Block Size	2^{-40}	2^{-20}	2^{-1}
64	2^{11}	2^{21}	2^{31}
128	2^{43}	2^{53}	2^{63}

CTR Mode

- A stream cipher mode that can encrypt arbitrary blocks of data (though specified to work 1 block at a time)
- Intended as a safer and more predictable version of OFB mode.
  - Instead of using prior keystream as the next cipher input, uses a counter.
  - The counter increments in some defined way.
  - The counter can not be allowed to repeat under the same key.
- Output of cipher is the keystream, and is XORed with the plaintext to create the ciphertext.

CTR Diagram



Source: NIST SP800-38A

CTR Notes

- Uses cipher in encrypt mode only
- Keystream is in no way affected by the plaintext
- Anytime there is a counter collision under the same key, the produced keystream is the same.
- In the instance of (Key,Counter) collision, *all future* plaintexts are revealed, and the corresponding past plaintexts.
- In the event of an error:
  - Bitwise equivalent changes occur in the corresponding plaintext.
  - Blocks after the altered block are unaffected.
  - An attacker can perform bit-level targeted modification to the plaintext.

Stream Ciphers: The Rosencrantz & Guildenstern Affair

- XOR desired message with plaintext message.
- XOR the result with the ciphertext.
- Decryption is now the desired message.

Plaintext
From: King Claud
ius To: The King
of England Plea
se help me kill
my nephew Hamlet
. Please send me
evidence with my
loyal chattel:
my men Rosencra
ntz & Guildenste
rn.

Stream Ciphers: The Rosencrantz & Guildenstern Affair

- XOR desired message with plaintext message.
- XOR the result with the ciphertext.
- Decryption is now the desired message.

Plaintext		Decrypted Plaintext
From: King Claud		From: King Claud
ius To: The King		ius To: The King
of England Plea		of England I am
se help me kill		responsible for
my nephew Hamlet	Attack →	killing my broth
. Please send me		er and taking his
evidence with my		wife. Hamlet is
loyal chattel:		a swell guy. Giv
my men Rosencra		e him an army to
ntz & Guildenste		depose me. Toodl
rn.		es. – Claudius

Subsection 3

Data Integrity Modes

CBC-MAC Mode

- An integrity mode, based on the CBC encryption mode.
- Set the IV to all 0s.
- Encrypt the data to be protected, and discard all ciphertext other than the last block.
- Optionally truncate this block of ciphertext.
- Send the data (possibly separately encrypted) and the CBC-MAC.

CBC-MAC Notes

- If the plaintext to be authenticated is not block aligned, the last block must be padded.
- There is message ambiguity unless this padding is unambiguous.
- CBC-MAC keys should be different than encryption keys (particularly when used for CBC encrypt mode!)
- Security in the case of fixed length messages, or messages that include the message length in the first block, is excellent.
- Other uses (no length specified, or the length in the last block) suffer from extension attacks.

A Note on Message Padding

- Some naïve padding schemes can lead to message ambiguity.
  - The MAC of a message that was padded and a message whose end happens to resemble padding should not be the same.
  - Simply appending all 0s or all 1s is ambiguous. Where does the message end?
  - One common system is appending a binary 1 followed by as many 0s as necessary
- For CBC-MAC, to prevent message ambiguity, one must do at least one of the following:
  - Force all messages to be same fixed length.
  - Prepend the message length (and reject messages of the incorrect length).
  - Always unambiguously pad, even for messages that are block aligned (*n.b.* block aligned messages have a full block of padding added).

An Example of the CBC-MAC Extension Attack

- The attacker can query a CBC-MAC oracle which operates using key k.
- The attacker requests the CBC-MAC of a one block message, m, from the oracle and obtains t_1.
- The attacker requests the CBC-MAC of the message t_1 from the oracle, obtaining t_2.
- The attacker then knows the CBC-MAC of the two block message $(m||0)$, namely t_2 (where $||$ denotes string concatenation.)

A More General CBC-MAC Extension Attack

- The attacker requests the CBC-MAC of the messages m and m', obtaining t and t', respectively.
- Denote the blocks making up m' as $m' = m'_1 || \cdots || m'_n$.
- The attacker then knows the MAC for a derived message, $m || \left(m'_1 \oplus t \right) || m'_2 || \cdots || m'_n$, namely t'.

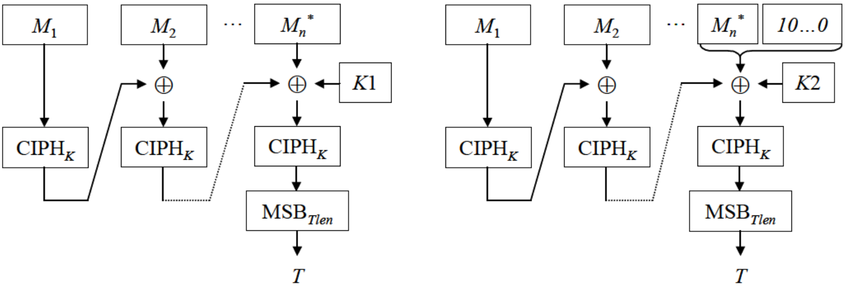
CBC-MAC and CBC Mode: Why Distinct Keys?

- As a general principle of cryptographic hygiene, keys should be used for only a single type of use:
  - A single mode of use
  - A single class of data (e.g., keys used to protect key management data should be different than keys used to protect user data.)
- Using the same key for both CBC-MAC and CBC encryption is useless.
  - So long as the attacker does not modify the final block of ciphertext, the CBC-MAC of the decrypted message will remain constant.

CMAC Mode

- A modern integrity mode, based on Phillip Rogaway's OMAC1
- Does not suffer from extension attacks
- Does not require encoding the length in the data, or a fixed data length.
- Pads partial last blocks unambiguously
- Uses one of two different subkeys, K1 and K2, for the final block, depending on the length of the last block.
  - Subkeys are derived by encrypting a block of 0s under K (+ some other processing).
- Allows truncating the resulting MAC output to 64 bits or longer.

CMAC Diagram



Source: NIST SP800-38B

Subsection 4

Combined Confidentiality / Integrity Modes

CCM Mode

- Provides data integrity (via CBC-MAC)
  - CBC-MAC is applied to the plaintext data.
  - Data length is prepended to data stream.
  - There is a provision for some data to be authenticated but not encrypted.
- Provides confidentiality (via CTR mode)
  - MAC is encrypted

Subsection 5

Block Cipher Modes of Operation Conclusion

What Did You Learn in School Today?

- Even idealized block ciphers are not magical.
- Some cipher modes are brittle.
- Most block cipher modes allow the attacker to make some targeted changes to the decrypted plaintext.
- Stream-cipher-like modes allow the attacker to make any desired changes to the decrypted plaintext.
- Detection of noise or attacker-induced corruption should not be dependent on message structure.
- Any confidentiality mode should be used in conjunction with some reasonable data integrity scheme.
- There are some modes that provide data integrity/authenticity.
- There are some modes that provide both confidentiality and data integrity/authenticity.

Block Cipher Example Outline

1. Introduction
2. Block Cipher Modes of Operation
3. Block Cipher Construction
   - DES
   - AES
   - DES/AES Conclusion

General Approaches

- Substitution
- Permutation
- Expansion
- Compression (de-expansion)
- “Math”
- Integration of Keying Material

One can (and should) combine these approaches. This is called a product cipher.

Subsection 1

DES

DES: History

- IBM designed several variants of a cipher called Lucifer.
- One of these variants (Feistel network, 64 bit block size, 64 bit key) was submitted to the National Bureau of Standards (NBS).
- The NSA worked with IBM to tune the algorithm:
  1. Reduced the key size to 56 bits.
  2. Changed the S-Boxes.

- DES was adopted in by the NBS in FIPS 46 in 1977, and then renewed as a standard in 1983, 1988, 1993, and 1999. FIPS 46-3 was withdrawn in 2005.
- The NSA changes were controversial, and many suspected that the NSA weakened the design.
- This induced (or was coincident with) the rise of an academic cryptologic research community.

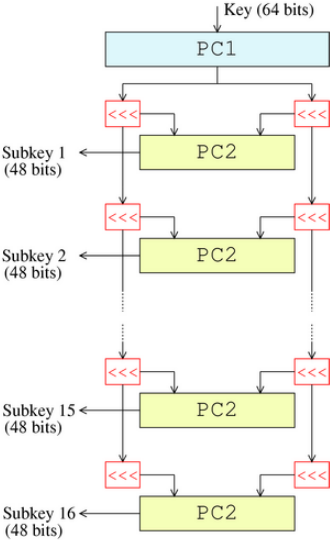
DES: History

- IBM designed several variants of a cipher called Lucifer.
- One of these variants (Feistel network, 64 bit block size, 64 bit key) was submitted to the National Bureau of Standards (NBS).
- The NSA worked with IBM to tune the algorithm:
  1. Reduced the key size to 56 bits.
  2. ~~Changed the S-Boxes.~~
  3. Don Coppersmith revealed that IBM developed the S-Boxes (and knew about differential cryptanalysis).
- DES was adopted in by the NBS in FIPS 46 in 1977, and then renewed as a standard in 1983, 1988, 1993, and 1999. FIPS 46-3 was withdrawn in 2005.
- The NSA changes were controversial, and many suspected that the NSA weakened the design.
- This induced (or was coincident with) the rise of an academic cryptologic research community.

DES: General Design

- 64 bit block size.
- 56 bit key (in a 64 bit block; LSB of each byte is optionally a parity bit to force odd parity).
- Cipher consists of a few distinct components:
  - Key Scheduler (establishes encrypt or decrypt)
  - The Initial Permutation (and its corresponding inverse, the Final Permutation)
  - Feistel network (16 rounds).

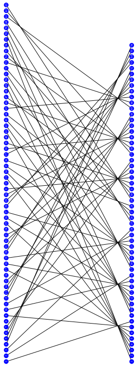
DES: Key Scheduler



Source: Matt Crypto via: Wikipedia

DES: PC1

			C_0			
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
			D_0			
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4



Source: SebDE via: Wikipedia

- Strip away the parity bits.
- No repeated or dropped key bits.
- Reorder key.
- Split the key into two 28 bit blocks, C_0 and D_0.

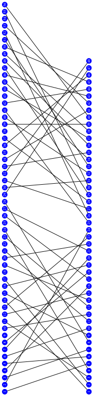
"<<<"

- $C_i = C_{i-1} <<< s_i$, $D_i = D_{i-1} <<< s_i$
- Circular shift left.
- Number of shifts depends on the round index:

	Round (i)															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
s_i	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

DES: PC2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32



Source: SebDE via: Wikipedia

- Select 48 bits.
- Drop bits 9, 18, 22, 25, 35, 38, 43, 54.
- Reorder key.

DES: Key Schedule Properties

- Outputs subkeys $K_1, \ldots, K_{16}$.
- Every bit of the key is in roughly 14 of the 16 subkeys.
- Ordering of subkeys establishes encrypt or decrypt mode of DES.

DES: Initial Permutation

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

- A perverse holdover from 1970s hardware design.
- Absolutely no security impact.
- Think: Eight 8-bit shift registers, fed by an 8-bit bus.

DES: Initial Permutation

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

- A perverse holdover from 1970s hardware design.
- Absolutely no security impact.
- Think: Eight 8-bit shift registers, fed by an 8-bit bus.
- :-(

DES: Initial Permutation

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

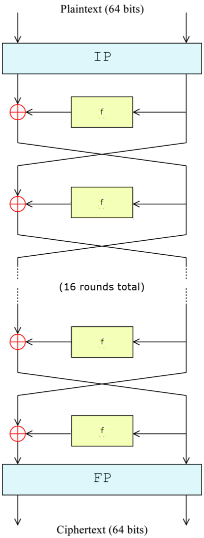
- A perverse holdover from 1970s hardware design.
- Absolutely no security impact.
- Think: Eight 8-bit shift registers, fed by an 8-bit bus.
- :-(
- On the positive side, it's very slow in software...

DES: Initial Permutation

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

- A perverse holdover from 1970s hardware design.
- Absolutely no security impact.
- Think: Eight 8-bit shift registers, fed by an 8-bit bus.
- :-(
- On the positive side, it's very slow in software...
- :-(:-(

DES: Feistel Network

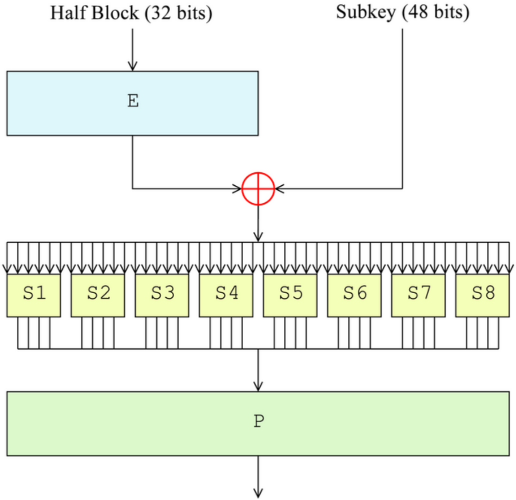


Source: Matt Crypto via: Wikipedia

DES: Feistel Network Notes

- 16 rounds total (16 invocations of the *f* function).
- Each of the subkeys from the key scheduler is used for one invocation of *f*.
- The current right half of the block is used as input for *f*.
- The output of *f* is XORed with the left half of the block.
- *f* need not be invertible.
- If *f* is linear, this network is fully linear (in $\mathbb{F}_2$).
- Last step does not exchange (so encrypted sides are “swapped”).
- Inverting (decrypting) only requires regenerating the same inputs for *f* (in reverse order). This is the encrypt process, but with subkeys reversed.

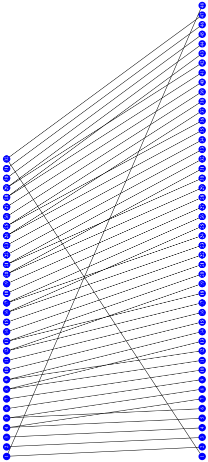
DES: *f* Function



Source: Matt Crypto via: Wikipedia

DES: *f* Function Expansion

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1



Source: SebDE via: Wikipedia

DES f Function Expansion Goals

- Expand the input into eight 6-bit blocks.
- The middle 4 bits of each 6-bit block contain the complete message.
- Two adjacent blocks share two "message bits" (S_1 and S_8 are considered adjacent.)

DES: S-Box

- The non-linear component of f.
- Substitution and compression.
  - Each S-Box takes in 6 bits and outputs 4 bits.
  - Each S-Box acts by applying one of 4 permutations on the middle 4 bits.
  - The particular permutation used is selected depending on the first and last input bit.
- Selection of the S-Box is of paramount importance for the security of DES.
- S-Box design was made to be maximally resistant to differential cryptanalysis (an attack publicly known in the late 80s).
- S-Box design was not made resistant to linear cryptanalysis (an attack publicly known in 1992).

Example S-Box

Input:

x_1	y_1	y_2	y_3	y_4	x_2

	S1															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Example Use of S-Box 1

Input:

1	0	1	1	1	0

	S1															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Output:

1	0	1	1

DES: S-Box Design Goals

S-1 No output bit of an S-box should be too close to a linear function of the input bits.

S-2 If two inputs to an S-box differ in exactly one bit, the outputs must differ in at least two bits.

S-3 If two inputs to an S-box differ in the two middle bits exactly, the outputs must differ in at least two bits.

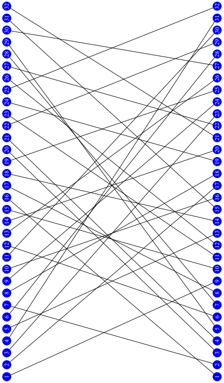
S-4 If two inputs to an S-box differ in their first two bits and are identical in their last two bits, the two outputs must not be the same.

S-5 For any nonzero 6-bit difference between inputs no more than eight of the 32 pairs of inputs exhibiting this difference may result in the same output difference.

S-6 Similar to (S-5), but with stronger restrictions in the case that there are three “active” S-Boxes.

DES: *f* Function Permutation

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25



Source: SebDE via: Wikipedia

DES: *f* Function Permutation Design Goals

- P-1 The four output bits from each S-box at round i are distributed so that two of them affect "middle bits" of S-boxes at round $i + 1$ (recall: the two middle bits of input to an S-box are not shared).
- P-2 The four output bits from each S-box affect six different S-boxes; no two affect the same S-box.
- P-3 For two S-boxes j, k, if an output bit from S_j affects a middle bit of S_k, then an output bit from S_k cannot affect a middle bit of S_j .

DES: Cryptographic Foibles

- $y = e_k(m) \Leftrightarrow \bar{y} = e_{\bar{k}}(\bar{m})$.
- If $K_i = K_j$ for all i, j, then k is a weak key.
  - Encryption and decryption functions are the same.
  - For each weak key, there are 2^{32} fixed points of e_k.
  - There are four weak keys.
- If $e_k = d_{k'}$ then (k, k') are a semi-weak key pair.
  - There are six pairs of semi-weak keys.
  - For each of the semi-weak key pairs, there are 2^{32} anti-fixed points $e_k(x) = \bar{x}$.

DES: Theoretical Attack Landscape

- Brute force
  - 2^{55} operations, with 1 known plaintext.
  - Negligible block storage requirement.
- Differential Cryptanalysis
  - Chosen plaintext attack.
  - 2^{47} operations, 2^{47} messages.
  - DES is optimized against Differential Cryptanalysis.
- Linear Cryptanalysis
  - Known plaintext attack.
  - Time / memory / success trade off
  - 2^{41} operations, 2^{43} messages (85% chance of success)
  - 2^{50} operations, 2^{38} messages (10% chance of success).
  - DES is not optimized against Linear Cryptanalysis.
- Only brute force is considered a viable attack in most settings.

DES: Attack History

Year	Notes	Cost	Runtime
1977	Hellman design	\$20M (est.)	1 day
1993	Wiener design	\$1M (est.)	7 hours
1997	DESCHALL	free (as in lemonade)	96 days
1998	EFF (“Deep Crack”)	\$250k	56 hours
2006–2008	COPACOBANA(s)	\$10k	7 days

DES: Current Use

- Two structural problems with using DES:
  - The block size is small.
  - The key size is small.
- The problem with key size can be addressed by iterating DES.
- DES is not a group, so iterated DES may be useful.
- By iterating DES (allowing distinct keys), there are provably at least 10^{2499} distinct permutations obtainable.

DES: Iterated DES

- Iterating increases the number of key bits, which sounds good.
- Double encryption is not helpful:
  - A meet in the middle attack gives only a doubling of computational security!
  - Encrypt known plaintext with all possible keys.
  - Decrypt associated ciphertext with all possible keys. Look for matches.
  - Information theory tells us that we need two distinct ciphertext/plaintext pairs to uniquely identify the keys.
  - This attack requires on average $2^{56.6}$ operations with 2^{56} storage.
- Triple encryption is more helpful than the above suggests.

$$\text{Input} \rightarrow e_{k_1} \rightarrow d_{k_2} \rightarrow e_{k_3} \rightarrow \text{Output}$$

- Traditionally, one uses Encrypt-Decrypt-Encrypt (EDE) for encryption.
- Decrypt-Encrypt-Decrypt (DED) is then used for decryption.
  - This is not interesting for DES!
  - The only difference between encrypt and decrypt mode is the order in which subkeys are used.
- There are three common modes of Triple-DES.
  - 3-Key Triple-DES (All keys are distinct)
  - 2-Key Triple-DES ($k_1 = k_3$, k_2 is distinct)
  - 1-Key Triple-DES ($k_1 = k_2 = k_3$)

DES: Triple-DES Security

- Three-key triple-DES can be attacked in 2^{112} operations; this attack requires 2^{56} blocks of storage.
- Two-key triple-DES has a time-storage tradeoff attack:
  - 2^{56} operations, 2^{56} storage required for attack, 2^{56} chosen plaintexts.
  - 2^{80} operations, 2^{40} storage required for attack, 2^{40} chosen plaintexts.
- One-key triple-DES is equivalent to single-DES.
- "Internal chaining" weakens the cipher a great deal, generally to the level of single-DES.

Subsection 2

AES

AES: The King is Dead, Long Live the King!

By 1997 it was clear that DES had some problems:

- The key length is too short for the modern computational environment.
- Triple-DES is very slow, particularly in software.
- The 64 bit block length leads to intrinsic limitations (birthday paradox problems).

AES: The Selection Process

- September 12, 1997: NIST (née NBS) solicited submissions of ciphers to replace DES.
  - Required a block size of 128 bits.
  - Required a selectable key size of 128, 192, or 256 bits.
- 15 ciphers were submitted
- Conferences were scheduled to present cryptanalysis on the candidates.
  - AES1, August 1998
  - AES2, March 1999
  - AES3, April 2000.
- In August 1999, 5 finalists were selected by NIST.
- October 2, 2000 NIST announced that they had chosen Rijndael.

AES: General Design

- 128 bit block size.
- 128, 192 or 256 bit key.
- An invocation of the Cipher consists of a few distinct components:
  1. Key Scheduler
  2. AddRoundKey
  3. $N_r - 1$ Rounds
  4. FinalRound

A Word on Finite Fields

- We are working in various finite fields.
- As a matter of practicality, we need to establish how to represent the field elements as binary strings.
- We'll express field elements as elements in $\mathbb{F}_2[x]/\langle m(x)\rangle$ where $m(x)$ is a degree 8 irreducible polynomial.
- Think: the set of polynomials of degree 7 or less, standard addition, reduce multiplication by the (irreducible) polynomial $m(x)$.
- We view bytes as elements of $\mathbb{F}_{2^8}$. (MSB: b_7. LSB: b_0)

$$b_7b_6b_5b_4b_3b_2b_1b_0 \leftrightsquigarrow \sum_{i=0}^{7} b_i x^i$$

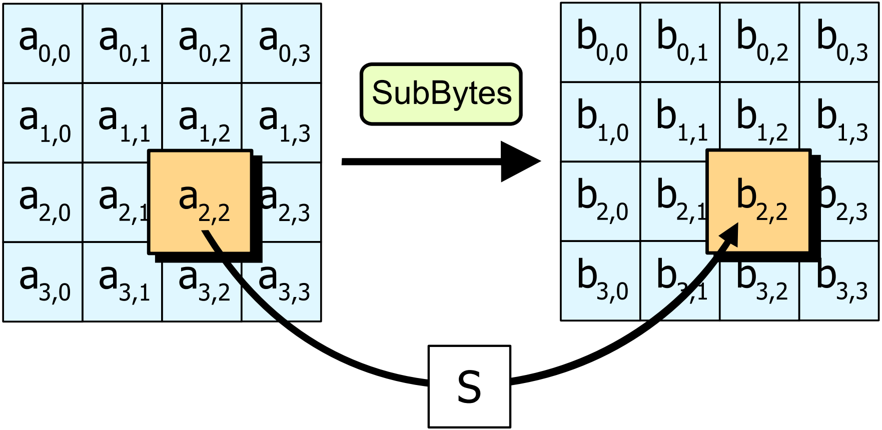
- For uniqueness of representation, we fix $m(x) = x^8 + x^4 + x^3 + x + 1$.

AES: Round

1. SubBytes
2. ShiftRows
3. MixColumns
4. AddRoundKey

The final round is the same, but without the MixColumns transform.

AES: SubBytes Transform



Source: Matt Crypto via: Wikipedia

AES: SubBytes Transform Specification

- The transform is $S = f \circ g$ where:
  1. $g : \mathbb{F}_{2^8} \to \mathbb{F}_{2^8}$

$$g(a) = \begin{cases} a^{-1} & a \neq 0 \\ 0 & a = 0 \end{cases}$$

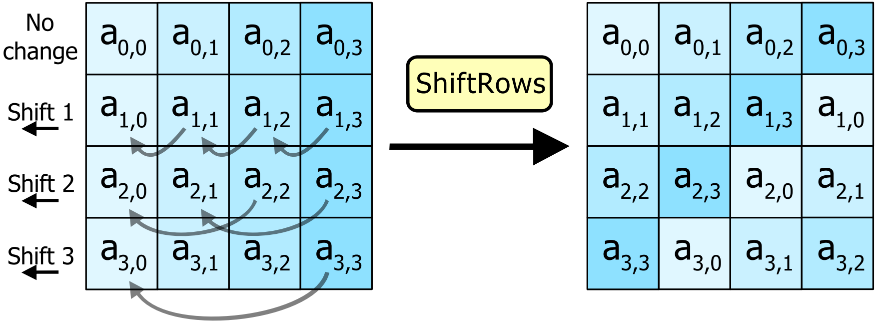
  2. $f : \mathbb{F}_2^8 \to \mathbb{F}_2^8$

$$f\left(\begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}\right) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

AES: SubBytes Notes

- Every byte has exactly the same substitution.
- This substitution is not key dependent.
- The inverse operation is non-linear.
- The affine transformation was selected so that the algebraic expression of S in $\mathbb{F}_{2^8}$ is complex.
- The affine transformation was selected so that there are no fixed or anti-fixed points.
- Both operations are clearly invertible; $S^{-1} = g^{-1} \circ f^{-1}$.

AES: ShiftRows Transform

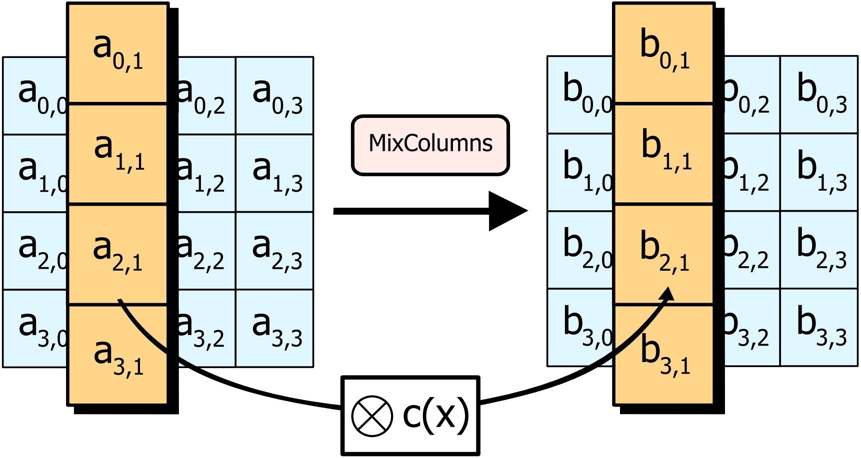


Source: Matt Crypto via: Wikipedia

AES: ShiftRows Transform Notes

- Cyclically shifts left the rows by different fixed amounts.
- Provides mixing of columns (diffusion).
- This is not key dependent.
- This helps protect against truncated differential and saturation attacks.
- The inverse operation just shifts the same amount in the opposite direction.

AES: MixColumns Transform



Source: Matt Crypto via: Wikipedia

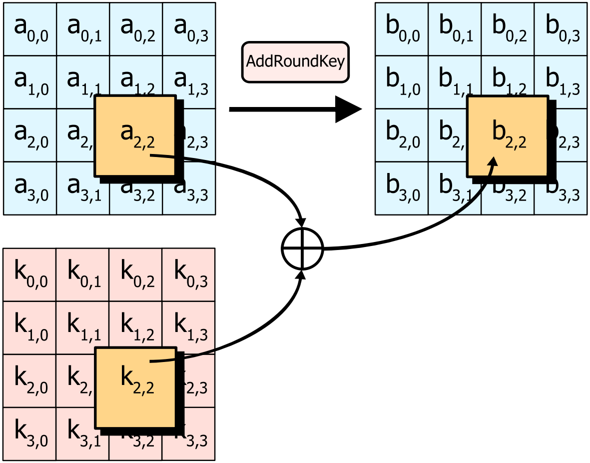
AES: MixColumns Transform Specification

- Each column is viewed as the coefficients of a polynomial (of degree less than 4).
- The row number of the column is the power of x.
- This polynomial is multiplied by the fixed polynomial $c(x) = \texttt{03}x^3 + \texttt{01}x^2 + \texttt{01}x + \texttt{02}$ and then reduced mod $x^4 + 1$.

AES: MixColumns Transform Notes

- This is linear over $\mathbb{F}_2$.
- This contributes to diffusion.
- $c(x)$ is coprime with $x^4 + 1$, so it has an inverse:
  $\texttt{0B}x^3 + \texttt{0D}x^2 + \texttt{09}x + \texttt{0E}$
- This is not key dependent.

AES: AddRoundKey Transform



Source: Matt Crypto via: Wikipedia

AES: AddRoundKey Transform Notes

- This combines the round key with the data block through XOR.
- This is key dependent.
- This transform is self-inverting.

AES: Number of Rounds

	Key size		
	128	192	256
Rounds	10	12	14

- Full diffusion is provided after 2 rounds.
- Best known attacks at time of design stopped working starting at 6 rounds.
- Inserted "safety margin" of one full diffusion step at start and end of cipher, gives the 10 round value.
- More rounds are used for longer keys because the number of rounds makes various "short cut" attacks harder (and they should be as hard as guessing the longer key).
- Longer keys give an attacker more power for known- and related-key attacks.

AES: Key Scheduler

- Expands the key from N_k 32-bit words to $N_r + 1$ round keys, each of which is 1 block (128 bits) long.
- First copies the provided key into the expanded key array.
- For later words, the ith word is the word one key length prior (the $(i - N_k)$th word) XORed with:
  - For words that are at a multiple of N_k, a processed version of the $(i - 1)$th word:
    1. Circularly rotated left one byte.
    2. S is applied to all bytes in the word.
    3. The first byte of the word is XORed with $(\texttt{02})^{i/N_k - 1}$.
  - When $N_k = 8$, if $i \equiv 4 \pmod 8$, the prior word after S has been applied to each byte.
  - Otherwise the prior word.

AES Security

- AES (with 256 bit keys) is the only non-classified block cipher that is approved to protect Top Secret information.
- AES was designed to perform well against all known attacks at the time, and had a "safety margin" to help with unknown attacks.
- In 2009, a related key attack was found on the full 192 and 256 key length AES with work factor requiring 2^{119} computations and plaintext / ciphertext pairs.
- In 2011, there was a full key recovery attack found which reduces security by at most 1.25 bits requiring 2^{80} ciphertext / plaintext pairs.

Subsection 3

DES/AES Conclusion

DES Summary

- DES has been attacked for nearly 40 years.
- Several serious attacks have been found, through they are largely impractical.
- The most practical attack is simple brute force, which is practical.
- (Three-Key) Triple-DES addresses the brute force attack, but is very slow.
- Some birthday paradox related problems with the block size remain.

AES Summary

- AES has been attacked for 14 years.
- No serious attacks have been found, though there are hints that some serious reduction in strength may be possible.
- The most practical attack is simple brute force, which is not practical.
- Birthday attacks are largely addressed by the large 128-bit block size.

Colophon

- The principal font is Evert Bloemsma's 2004 humanist san-serif font Legato. This font is designed to be exquisitely readable, and is a significant departure from the highly geometric forms that dominate most san-serif fonts. Legato was Evert Bloemsma's final font prior to his untimely death at the age of 46.
- Math symbols are typeset using the MathTime Professional II (MTPro2) fonts, a font package released in 2006 by the great mathematical expositor Michael Spivak.
- The URLs are typeset in Luc(as) de Groot's 2005 Consolas, a monospace font with excellent readability.