Symmetric Cryptography

Bernardo David

(Original Slides by Rosario Giustolisi)

Early Course Evaluation

- What do you like about the course?
- What do you dislike about the course?
- Suggestions for the lectures?
- Suggestions for the exercise sessions?
- Any other comments?

Review

- Cyclic Groups (revise for next lecture)
- Attacks on the network stack: no confidentiality and integrity!
- Denial of Service attacks: undermining availability!
- Firewalls: still no confidentiality and integrity!

Plan

- Symmetric Encryption for **Confidentiality**
  - Caesar’s cipher
  - Perfect secrecy
  - Block cipher
- Hash and MAC for **Integrity and Authenticity**
  - Hash definitions
  - MAC
  - Applications

Motivation

- Protect data confidentiality, integrity and availability on insecure networks.
- Based on strong mathematical foundations: proven to be secure instead of conjectured.
- We use cryptography every day to access web sites and services securely.
- Our goal: understand the main cryptographic tools and how they are used in practice.

Symmetric Encryption

confidentiality

Source: vsco.co

Encryption - Fundamentals

- Science of transforming a given string into a different one that is semantically equivalent (encryption) because the latter can be transformed back to the original one (decryption)

- An algorithm $c = \mathcal{E}(m,k)$ to encrypt a plaintext m producing a ciphertext c
- An algorithm $m = \mathcal{D}(c,k')$ to decrypt a ciphertext c producing a plaintext m
- It is not necessarily $k = k'$
  - k = *KeyGen()* being a random string

- Symmetric vs Asymmetric (aka Public key)

Symmetric Encryption

Definition

- The same key as the one that was used to create a ciphertext by encrypting a plaintext shall be used to decrypt the ciphertext back as the plaintext

- Goal: confidentiality

- Correctness: $\mathcal{D}(\mathcal{E}(m,k),k)$ = m
- Security: ∀k’. k’ ≠ k ⟶ $\mathcal{D}(\mathcal{E}(m,k),k')$ ≠ m

- Examples:
  - Caesar’s cipher, DES, 3DES, AES
  - Typical key length: 128/256 bits
  - Good performance

Encryption Question

Select the **correct** option:

a) Encryption guarantees message authenticity.

b) Symmetric encryption can be used by two parties who each know a different encryption key.

c) Symmetric encryption protects confidentiality of encrypted messages.

d) Dolev—Yao adversaries can read messages encrypted under a symmetric encryption scheme.

Caesar cipher

- k = *KeyGen()*
- $c = \mathcal{E}(m,k)$
- $m = \mathcal{D}(c,k)$

Source: pinterest

Caesar cipher

- $k = \mathit{KeyGen}()$: *rotation of the wheel*
  - E.g. A->F, then B->G, C->H, F->K etc.
- $c = \mathcal{E}(m,k)$: look at the inner wheel
- $m = \mathcal{D}(c,k)$: look at the outer wheel

Source: pinterest

Caesar cipher

- Example

- ABCDEFGHIJKLMNOPQRSTUVWXYZ
- A->N, B->O, C->P, …, N->A, O->B, etc. ← *key : shift(13)*

- $\mathcal{E}$(hamburgers the cornerstone of any nutritious breakfast, *shift(13)*) =

  unzohetref gur pbearefgbar bs nal ahgevgvbhf oernxsnfg

Caesar cipher

Key-space is too small: unzohetref gur pbearefgbar bs nal ahgevgvbhf oernxsnfg

+1	tmyngdsqde ftq oadzqdefazq ar mzk zgfdufuage ndqmwrmef
+2	slxmfcrpcd esp nzcypcdezyp zq lyj yfectetzfd mcplvqlde
+3	rkwlebqobc dro mybxobcdyxo yp kxi xedbsdsyec lbokupkcd
+4	qjvkdapnab cqn lxawnabcxwn xo jwh wdcarcrxdb kanjtojbc
+5	piujczomza bpm kwzvmzabwvm wn ivg vcbzqbqwca jzmisniab
+6	ohtibynlyz aol jvyulyzavul vm huf ubaypapvbz iylhrmhza
+7	ngshaxmkxy znk iuxtkxyzutk ul gte tazxozouay hxkgqlgyz
+8	mfrgzwljwx ymj htwsjwxytsj tk fsd szywnyntzx gwjfpkfxy
+9	leqfyvkivw xli gsvrivwxsri sj erc ryxvmxmsyw fvieojewx
+10	kdpexujhuv wkh fruqhuvwrqh ri dqb qxwulwlrxv euhdnidvw
+11	jcodwtigtu vjg eqtpgtuvqpg qh cpa pwvtkvkqwu dtgcmhcuv
+12	ibncvshfst uif dpsofstupof pg boz ovusjujpvt csfblgbtu
+13	hamburgers the cornerstone of any nutritious breakfast
+14	gzlatqfdqr sgd bnqmdqrsnmd ne zmx mtsqhshntr aqdzjezrs
+15	fykzspecpq rfc amplcpqrmlc md ylw lsrpgrgmsq zpcyidyqr
+16	exjyrodbop qeb zlokbopqlkb lc xkv krqofqflrp yobxhcxpq
+17	dwixqncano pda yknjanopkja kb wju jqpnepekqo xnawgbwop
+18	cvhwpmbzmn ocz xjmizmnojiz ja vit ipomdodjpn wmzvfavno
+19	bugvolaylm nby wilhylmnihy iz uhs honlcnciom vlyuezumn
+20	atfunkzxkl max vhkgxklmhgx hy tgr gnmkbmbhnl ukxtdytlm
+21	zsetmjywjk lzw ugjfwjklgfw gx sfq fmljalagmk tjwscxskl
+22	yrdslixvij kyv tfievijkfev fw rep elkizkzflj sivrbwrjk
+23	xqcrkhwuhi jxu sehduhijedu ev qdo dkjhyjyeki rhuqavqij
+24	wpbqjgvtgh iwt rdgctghidct du pcn cjigxixdjh qgtpzuphi
+25	voapifusfg hvs qcfbsfghcbs ct obm bihfwhwcig pfsoytogh

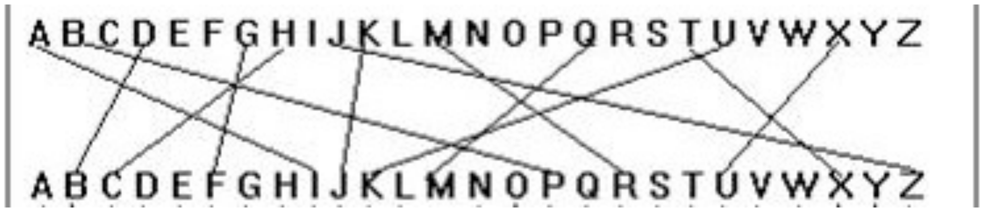
Caesar cipher – Brute force

Key-space is too small: unzohetref gur pbearefgbar bs nal ahgevgvbhf oernxsnfg

+1	tmyngdsqde ftq oadzqdefazq ar mzk zgfdufuage ndqmwrmef
+2	slxmfcrpcd esp nzcypcdezyp zq lyj yfectetzfd mcplvqlde
+3	rkwlebqobc dro mybxobcdyxo yp kxi xedbsdsyec lbokupkcd
+4	qjvkdapnab cqn lxawnabcxwn xo jwh wdcarcrxdb kanjtojbc
+5	piujczomza bpm kwzvmzabwvm wn ivg vcbzqbqwca jzmisniab
+6	ohtibynlyz aol jvyulyzavul vm huf ubaypapvbz iylhrmhza
+7	ngshaxmkxy znk iuxtkxyzutk ul gte tazxozouay hxkgqlgyz
+8	mfrgzwljwx ymj htwsjwxytsj tk fsd szywnyntzx gwjfpkfxy
+9	leqfyvkivw xli gsvrivwxsri sj erc ryxvmxmsyw fvieojewx
+10	kdpexujhuv wkh fruqhuvwrqh ri dqb qxwulwlrxv euhdnidvw
+11	jcodwtigtu vjg eqtpgtuvqpg qh cpa pwvtkvkqwu dtgcmhcuv
+12	ibncvshfst uif dpsofstupof pg boz ovusjujpvt csfblgbtu
+13	**hamburgers the cornerstone of any nutritious breakfast**
+14	gzlatqfdqr sgd bnqmdqrsnmd ne zmx mtsqhshntr aqdzjezrs
+15	fykzspecpq rfc amplcpqrmlc md ylw lsrpgrgmsq zpcyidyqr
+16	exjyrodbop qeb zlokbopqlkb lc xkv krqofqflrp yobxhcxpq
+17	dwixqncano pda yknjanopkja kb wju jqpnepekqo xnawgbwop
+18	cvhwpmbzmn ocz xjmizmnojiz ja vit ipomdodjpn wmzvfavno
+19	bugvolaylm nby wilhylmnihy iz uhs honlcnciom vlyuezumn
+20	atfunkzxkl max vhkgxklmhgx hy tgr gnmkbmbhnl ukxtdytlm
+21	zsetmjywjk lzw ugjfwjklgfw gx sfq fmljalagmk tjwscxskl
+22	yrdslixvij kyv tfievijkfev fw rep elkizkzflj sivrbwrjk
+23	xqcrkhwuhi jxu sehduhijedu ev qdo dkjhyjyeki rhuqavqij
+24	wpbqjgvtgh iwt rdgctghidct du pcn cjigxixdjh qgtpzuphi
+25	voapifusfg hvs qcfbsfghcbs ct obm bihfwhwcig pfsoytogh

Mono-alphabetic substitution

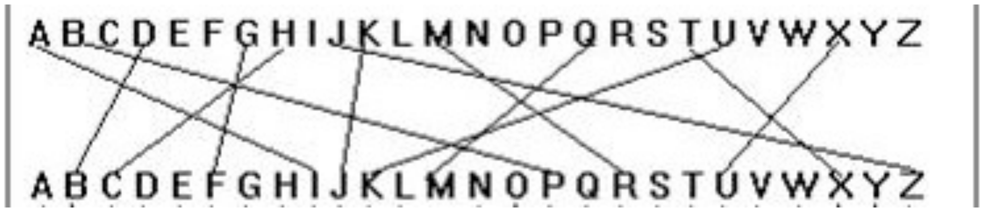
- Idea: Let's do a random permutation
- k = *KeyGen()*
  - E.g. A->N, B->F, C->S, …
- $c = \mathcal{E}(m,k)$
- $m = \mathcal{D}(c,k)$



- What about the key space?

Mono-alphabetic substitution

- Idea: Let's do a random permutation
- k = *KeyGen()*
  - E.g. A->N, B->F, C->S, …
- $c = \mathcal{E}(m,k)$
- $m = \mathcal{D}(c,k)$



- What about the key space?
- $26!=26*25*24*23*\ldots*1 > 4*10^{26}$

Mono-alphabetic substitution

VGUVGOGUZLWGVIGOGUCOLRNFUTCOZQLVULNUQIGUGFHGULBUQIGUFGZGOQUVIGNUQIGFORHZUTGHCNU
QLUQCYGUILMF3UKUOGWGWTGOUZCXKNHUZLWGQIKNHUMKYGU"KUBGGMUCTKQUMKHIQIGCFGF;UWCXT
GUXLRUZILRMFUFOKSG333U3”UCNFUZRFFGNMXUQIGOGUVCZUCUQGOOKTMGUOLCOUCMMUCOLRNFURZU
CNFUQIGUZYXUVCZUBRMMULBUVICQUMLLYGFUMKYGUIRHGUTCQZ2UCMMUZVLLJKNHUCNFUZDOGGDIKNH
UCNFUFKSKNHUCOLRNFUQIGUDCO2UVIKDIUVCZUHLKNHUCTLRQUCUIRNFOGFUWKMGZUCNUILROUVKQIUQI
GUQLJUFLVNUQLUMCZUSGHCZ3U

Mono-alphabetic substitution

VGUVGOGUZLWGVIGOGUCOLRNFUTCOZQLVULNUQIGUGFHGULBUQIGUFGZGOQUVIGNUQIGFORHZUTGHCNU
QLUQCYGUILMF3UKUOGWGWTGOUZCXKNHUZLWGQIKNHUMKYGU"KUBGGMUCTKQUMKHIQIGCFGF;UWCXT
GUXLRUZILRMFUFOKSG333U3”UCNFUZRFFGNMXUQIGOGUVCZUCUQGOOKTMGUOLCOUCMMUCOLRNFURZU
CNFUQIGUZYXUVCZUBRMMULBUVICQUMLLYGFUMKYGUIRHGUTCQZ2UCMMUZVLLJKNHUCNFUZDOGGDIKNH
UCNFUFKSKNHUCOLRNFUQIGUDCO2UVIKDIUVCZUHLKNHUCTLRQUCUIRNFOGFUWKMGZUCNUILROUVKQIUQI
GUQLJUFLVNUQLUMCZUSGHCZ3U

Symbols by frequency (English): ‘ ‘ E T A O I N S R H D L U C M F Y W G P B V K X Q J Z

Mono-alphabetic substitution

VGUVGOGUZLWGVIGOGUCOLRNFUTCOZQLVULNUQIGUGFHGULBUQIGUFGZGOQUVIGNUQIGFORHZUTGHCNU
QLUQCYGUILMF3UKUOGWGWTGOUZCXKNHUZLWGQIKNHUMKYGU"KUBGGMUCTKQUMKHIQIGCFGF;UWCXT
GUXLRUZILRMFUFOKSG333U3”UCNFUZRFFGNMXUQIGOGUVCZUCUQGOOKTMGUOLCOUCMMUCOLRNFURZU
CNFUQIGUZYXUVCZUBRMMULBUVICQUMLLYGFUMKYGUIRHGUTCQZ2UCMMUZVLLJKNHUCNFUZDOGGDIKNH
UCNFUFKSKNHUCOLRNFUQIGUDCO2UVIKDIUVCZUHLKNHUCTLRQUCUIRNFOGFUWKMGZUCNUILROUVKQIUQI
GUQLJUFLVNUQLUMCZUSGHCZ3U

Symbols by frequency (English): ‘ ‘ E T A O I N S R H D L U C M F Y W G P B V K X Q J Z

Symbols by frequency (cipher): U G Q C L K N Z I O F M R V D 3 H W X n B T J " 2 Y S

Mono-alphabetic substitution

VGUVGOGUZLWGVIGOGUCOLRNFUTCOZQLVULNUQIGUGFHGULBUQIGUFGZGOQUVIGNUQIGFORHZUTGHCNU
QLUQCYGUILMF3UKUOGWGWTGOUZCXKNHUZLWGQIKNHUMKYGU"KUBGGMUCTKQUMKHIQIGCFGF;UWCXT
GUXLRUZILRMFUFOKSG333U3”UCNFUZRFFGNMXUQIGOGUVCZUCUQGOOKTMGUOLCOUCMMUCOLRNFURZU
CNFUQIGUZYXUVCZUBRMMULBUVICQUMLLYGFUMKYGUIRHGUTCQZ2UCMMUZVLLJKNHUCNFUZDOGGDIKNH
UCNFUFKSKNHUCOLRNFUQIGUDCO2UVIKDIUVCZUHLKNHUCTLRQUCUIRNFOGFUWKMGZUCNUILROUVKQIUQI
GUQLJUFLVNUQLUMCZUSGHCZ3U

Symbols by frequency (English): ‘ ‘ E T A O I N S R H D L U C M F Y W G P B V K X Q J Z

Symbols by frequency (cipher): U G Q C L K N Z I O F M R V D 3 H W X n B T J " 2 Y S

CE CEHE SOWECREHE AHOUND VAHSTOC ON TRE EDYE OB TRE DESEHT CREN TRE DHUYS VEYAN TO TAJE
ROLDF I HEWEWVEH SAGINY SOWETRINY LIJE XI BEEL A VIT LIYRTREADED; WAGVE GOU SROULD DHIZEF F
F FX AND SUDDENLG TREHE CAS A TEHHIVLE HOAH ALL AHOUND US AND TRE SJG CAS BULL OB CRAT
LOOJED LIJE RUYE VATSQ ALL SCOOKINY AND SMHEEMRINY AND DIZINY AHOUND TRE MAHQ CRIMR CAS
YOINY AVOUT A RUNDHED WILES AN ROUH CITR TRE TOK DOCN TO LAS ZEYASF

Mono-alphabetic substitution

VGUVGOGUZLWGVIGOGUCOLRNFUTCOZQLVULNUQIGUGFHGULBUQIGUFGZGOQUVIGNUQIGFORHZUTGHCNU
QLUQCYGUILMF3UKUOGWGWTGOUZCXKNHUZLWGQIKNHUMKYGU"KUBGGMUCTKQUMKHIQIGCFGF;UWCXT
GUXLRUZILRMFUFOKSG333U3”UCNFUZRFFGNMXUQIGOGUVCZUCUQGOOKTMGUOLCOUCMMUCOLRNFURZU
CNFUQIGUZYXUVCZUBRMMULBUVICQUMLLYGFUMKYGUIRHGUTCQZ2UCMMUZVLLJKNHUCNFUZDOGGDIKNH
UCNFUFKSKNHUCOLRNFUQIGUDCO2UVIKDIUVCZUHLKNHUCTLRQUCUIRNFOGFUWKMGZUCNUILROUVKQIUQI
GUQLJUFLVNUQLUMCZUSGHCZ3U

Symbols by frequency (English): ‘ ’ E T A O I N S R H D L U C M F Y W G P B V K X Q J Z

Symbols by frequency (cipher): U G Q C L K N Z I O F M R V D 3 H W X n B T J " 2 Y S

CE CEHE SOWECREHE AHOUND VAHSTOC ON TRE EDYE OB TRE DESEHT CREN TRE DHUYS VEYAN TO TAJE ROLDF I HEWEWVEH SAGINY SOWETRINY LIJE XI BEEL A VIT LIYRTREADED; WAGVE GOU SROULD DHIZEF F F FX AND SUDDENLG TREHE CAS A TEHHIVLE HOAH ALL AHOUND US AND TRE SJG CAS BULL OB CRAT LOOJED LIJE RUYE VATSQ ALL SCOOKINY AND SMHEEMRINY AND DIZINY AHOUND TRE MAHQ CRIMR CAS YOINY AVOUT A RUNDHED WILES AN ROUH CITR TRE TOK DOCN TO LAS ZEYASF

Mono-alphabetic substitution

VGUVGOGUZLWGVIGOGUCOLRNFUTCOZQLVULNUQIGUGFHGULBUQIGUFGZGOQUVIGNUQIGFORHZUTGHCNU
QLUQCYGUILMF3UKUOGWGWTGOUZCXKNHUZLWGQIKNHUMKYGU"KUBGGMUCTKQUMKHIQIGCFGF;UWCXT
GUXLRUZILRMFUFOKSG333U3”UCNFUZRFFGNMXUQIGOGUVCZUCUQGOOKTMGUOLCOUCMMUCOLRNFURZU
CNFUQIGUZYXUVCZUBRMMULBUVICQUMLLYGFUMKYGUIRHGUTCQZ2UCMMUZVLLJKNHUCNFUZDOGGDIKNH
UCNFUFKSKNHUCOLRNFUQIGUDCO2UVIKDIUVCZUHLKNHUCTLRQUCUIRNFOGFUWKMGZUCNUILROUVKQIUQI
GUQLJUFLVNUQLUMCZUSGHCZ3U

Symbols by frequency (English): ‘ ‘ E T A O I N S R H D L U C **M** F Y W **G** P B V K X Q J Z

Symbols by frequency (cipher): U G Q C L K N Z I O F M R V D 3 H W X n B T J " 2 Y S

CE CERE SOWECHEHE AHOUND VAHSTOC ON THE EDYE OB THE DESEHT CHEN THE DHUYS VEYAN TO TAJE HOLDF I HEWEWVEH SAGINY SOWETHINY LIJE XI BEEL A VIT LIYHTHEADED; WAGVE GOU SHOULD DHIZEF F F FX AND SUDDENLG THEHE CAS A TEHHIVLE HOAH ALL AHOUND US AND THE SJG CAS BULL OB CHAT LOOJED LIJE HUYE VATSQ ALL SCOOKINY AND SMHEEMHINY AND DIZINY AHOUND THE MAHQ CHIMH CAS YOINY AVOUT A HUNDHED WILES AN HOUH CITH THE TOK DOCN TO LAS ZEYASF

Can you finish decrypting?

VGUVGOGUZLWGVIGOGUCOLRNFUTCOZQLVULNUQIGUGFHGULBUQIGUFGZGOQUVIGNUQIGFORHZUTGHCNU
QLUQCYGUILMF3UKUOGWGWTGOUZCXKNHUZLWGQIKNHUMKYGU"KUBGGMUCTKQUMKHIQIGCFGF;UWCXT
GUXLRUZILRMFUFOKSG333U3”UCNFUZRFFGNMXUQIGOGUVCZUCUQGOOKTMGUOLCOUCMMUCOLRNFURZU
CNFUQIGUZYXUVCZUBRMMULBUVICQUMLLYGFUMKYGUIRHGUTCQZ2UCMMUZVLLJKNHUCNFUZDOGGDIKNH
UCNFUFKSKNHUCOLRNFUQIGUDCO2UVIKDIUVCZUHLKNHUCTLRQUCUIRNFOGFUWKMGZUCNUILROUVKQIUQI
GUQLJUFLVNUQLUMCZUSGHCZ3U

Symbols by frequency (English): ‘ ‘ E T A O I N S R H D L U C **M** F Y W **G** P B V K X Q J Z

Symbols by frequency (cipher): U G Q C L K N Z I O F M R V D 3 H W X n B T J " 2 Y S

CE CERE SOWECHEHE AHOUND VAHSTOC ON THE EDYE OB THE DESEHT CHEN THE DHUYS VEYAN TO TAJE HOLDF I HEWEWVEH SAGINY SOWETHINY LIJE XI BEEL A VIT LIYHTHEADED; WAGVE GOU SHOULD DHIZEF F F FX AND SUDDENLG THEHE CAS A TEHHIVLE HOAH ALL AHOUND US AND THE SJG CAS BULL OB CHAT LOOJED LIJE HUYE VATSQ ALL SCOOKINY AND SMHEEMHINY AND DIZINY AHOUND THE MAHQ CHIMH CAS YOINY AVOUT A HUNDHED WILES AN HOUH CITH THE TOK DOCN TO LAS ZEYASF

Mono-alphabetic substitution

VGUVGOGUZLWGVIGOGUCOLRNFUTCOZQLVULNUQIGUGFHGULBUQIGUFGZGOQUVIGNUQIGFORHZUTGHCNU
QLUQCYGUILMF3UKUOGWGWTGOUZCXKNHUZLWGQIKNHUMKYGU"KUBGGMUCTKQUMKHIQIGCFGF;UWCXT
GUXLRUZILRMFUFOKSG333U3”UCNFUZRFFGNMXUQIGOGUVCZUCUQGOOKTMGUOLCOUCMMUCOLRNFURZU
CNFUQIGUZYXUVCZUBRMMULBUVICQUMLLYGFUMKYGUIRHGUTCQZ2UCMMUZVLLJKNHUCNFUZDOGGDIKNH
UCNFUFKSKNHUCOLRNFUQIGUDCO2UVIKDIUVCZUHLKNHUCTLRQUCUIRNFOGFUWKMGZUCNUILROUVKQIUQI
GUQLJUFLVNUQLUMCZUSGHCZ3U

Symbols by frequency (English): ‘ ’ E T A O I N S R H D L U C M F Y W G P B V K X Q J Z

Symbols by frequency (cipher): U G Q C L K N Z I O F M R V D 3 H W X n B T J " 2 Y S

WE WERE SOMEWHERE AROUND BARSTOW ON THE EDGE OF THE DESERT WHEN THE DRUGS BEGAN TO TAKE HOLD. I REMEMBER SAYING SOMETHING LIKE "I FEEL A BIT LIGHTHEADED; MAYBE YOU SHOULD DRIVE. . . ." AND SUDDENLY THERE WAS A TERRIBLE ROAR ALL AROUND US AND THE SKY WAS FULL OF WHAT LOOKED LIKE HUGE BATS, ALL SWOOPING AND SCREECHING AND DIVING AROUND THE CAR, WHICH WAS GOING ABOUT A HUNDRED MILES AN HOUR WITH THE TOP DOWN TO LAS VEGAS.

Perfect Secrecy

- Vernam cipher or One-time pad
- Idea: use the properties of XOR ($\oplus$) to encrypt and decrypt

- k = *KeyGen() : random*
- c= $\mathcal{E}(m,k) = m \oplus \text{k}$
- m = $\mathcal{D}(c,k) = m \oplus \text{k} \oplus \text{k} = \text{m}$

Source: Bansky

Perfect Secrecy

- Vernam cipher or One-time pad
- Idea: use the properties of XOR ($\oplus$) to encrypt and decrypt

- k = *KeyGen*() : *random*
- c= $\mathcal{E}(m,k) = m \oplus \mathrm{k}$
- m = $\mathcal{D}(c,k) = m \oplus \mathrm{k} \oplus \mathrm{k} = \mathrm{m}$

- 010101110101110101010 ← k
- $\mathcal{E}$(110101000011110101 1, k) = 1101010000111101011 $\oplus$

  0101011101011101010

  1000001101100000001

Perfect Secrecy

- Knowing the ciphertext tells you nothing about the message
  - For each ciphertext exists a key that maps to *any* plaintext
- Problem 1: key length = message length (impossible to have smaller keys)
  - The encryption of a 500GB hard drive would require 500GB RAM!
- Problem 2: the key can be used only once. Why?

Perfect Secrecy Question

Select the **correct** option:

a) A one-time pad key is agreed upon once and used many times.

b) A one-time pad is secure even against adversaries with unlimited computational power.

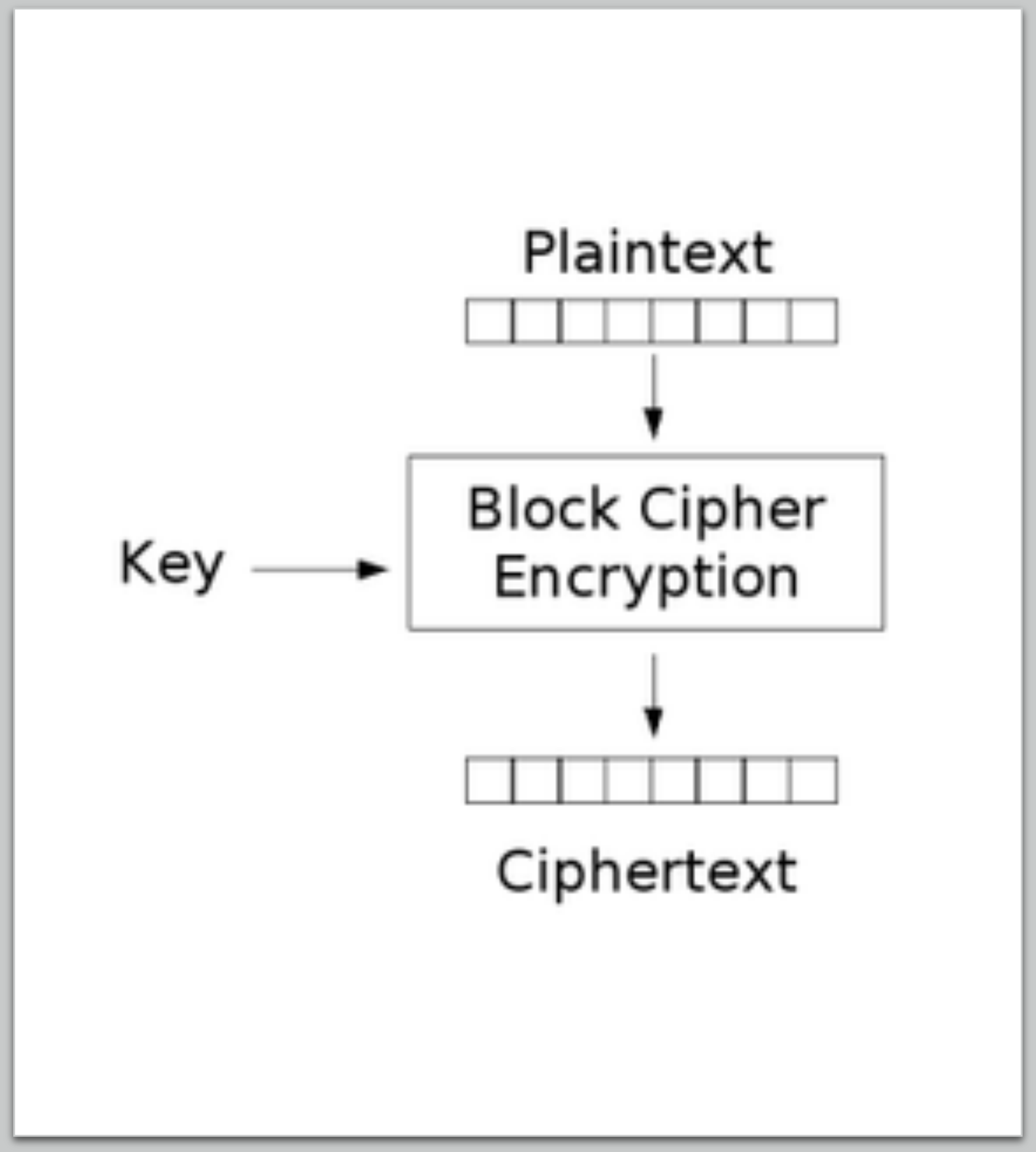
c) The length of the key in a one-time pad is independent from the message length.

d) One-time pads guarantee message integrity.

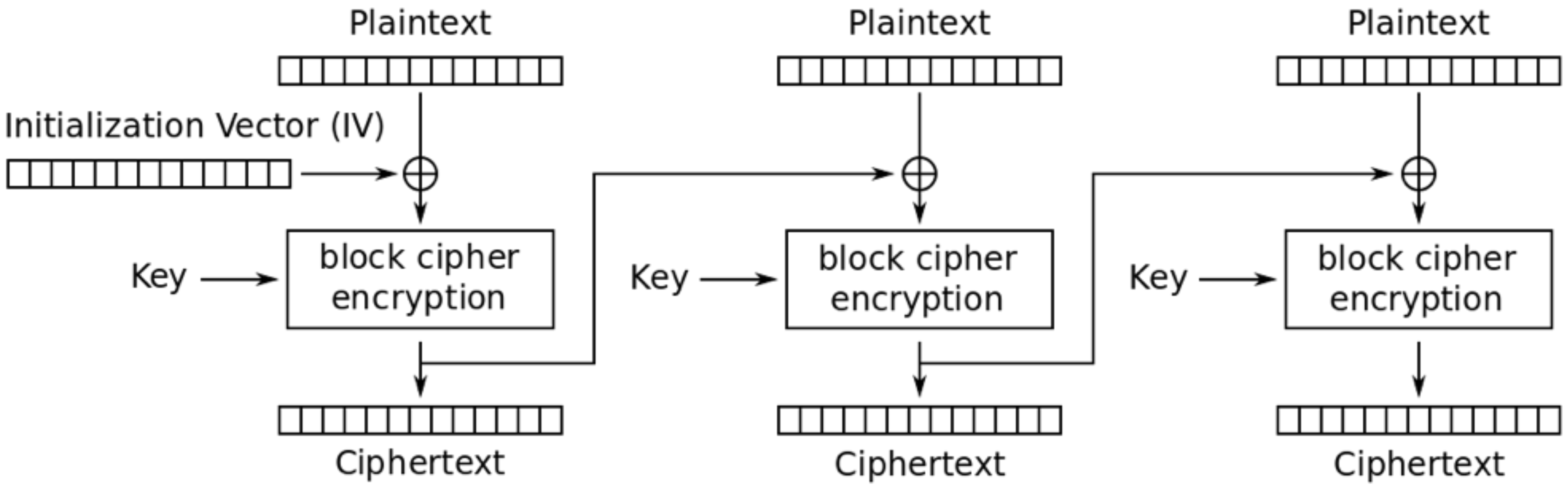
Block Cipher

- Problem 1: key length = message length
  - Idea: agree on a short key and generate fixed-length permutations from the key
- Problem 2: the key can be used only once.
  - Idea: use a random value on each encryption (aka initialisation vector)

- Multiple variants of block cipher exists



Cipher Block Chaining



Cipher Block Chaining (CBC) mode encryption

Block Cipher

- Security: A block cipher is secure if it is a good *pseudorandom permutation function*.

Pseudorandom permutation (very informal)

- The output of a secure cipher cannot be distinguished from a random permutation

How can we build a secure cipher?

Confusion

Make the connection between ciphertext and key as complicated as possible

Diffusion

Flipping a single bit of the plaintext (statistically) produces a flipping of half of the bits in the ciphertext

Block Cipher Question

Select the **incorrect** option:

a) It is possible to encrypt messages of any size with one call to a block cipher.

b) Confusion and Diffusion layers in block ciphers make the ciphertext highly uncorrelated with the message.

c) The key length in block ciphers used in CBC mode has no relation to the message length.

d) Block ciphers do not guarantee message authenticity.

AES

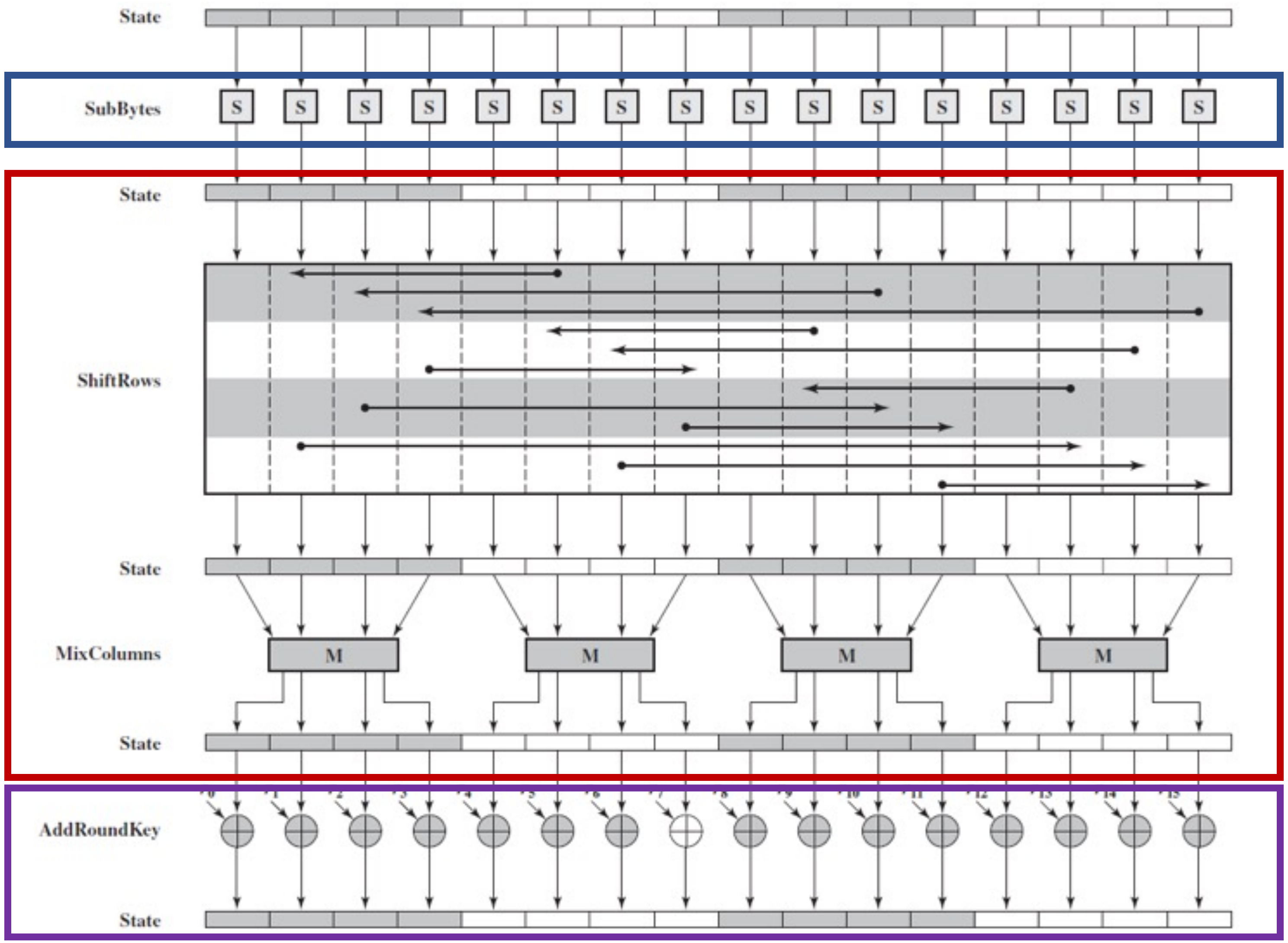
- Advanced Encryption Standard
  - State-of-the-art symmetric encryption algorithm
- Block length= 128 bits
  - Messages that are not multiple of block length are *padded.*
- Key length= 128, 192, or 256 bits
- Number of rounds= 10, 12, or 14
  - Each round consists of *layers*
- NSA classifies as TOP SECRET AES192 and AES256
  - No practical attack known on AES, when **correctly implemented**
- **We focus on AES128**

AES

Byte Substitution layer (S-Box)

Diffusion layer

Key Addition layer



AES

$AES(K,M)$ // $|M|$=128 and $|K|$=128

$(K_0,\ldots,K_{10})$<- KeySchedule(K) // $|K_i|$=128

$s \leftarrow M \oplus K_0$

for r=1 to 10

Byte Substitution layer (S-Box) — $s \leftarrow S(s)$

Diffusion layer — $s \leftarrow \text{ShiftRows}(s)$

if $r \leq 9$ then

$s \leftarrow \text{MixCols}(s)$

Key Addition layer — $s \leftarrow s \oplus K_r$

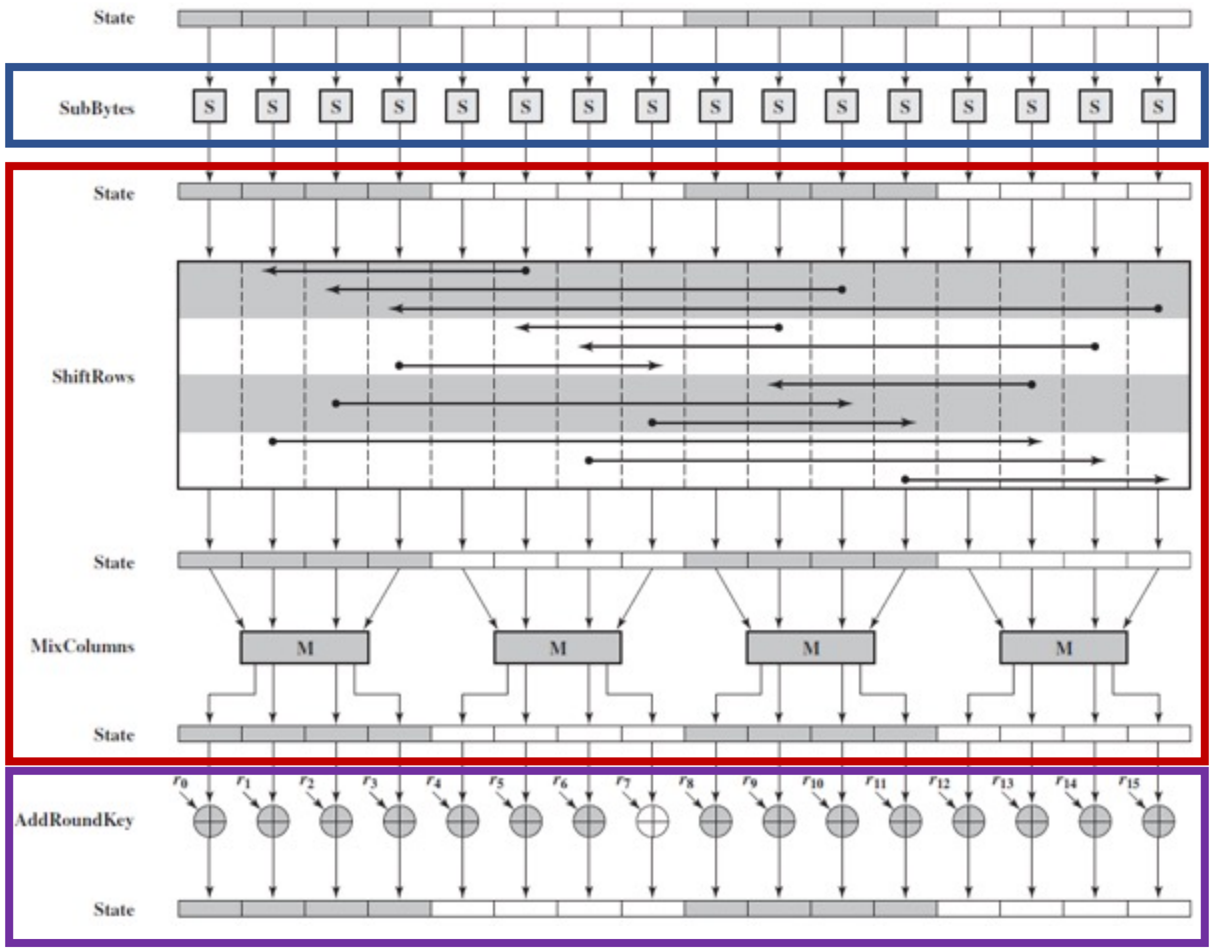
Return s

The state s is arranged in a 4x4 matrix

s

s_{00}	s_{10}	s_{20}	s_{30}
s_{01}	s_{11}	s_{21}	s_{31}
s_{02}	s_{12}	s_{22}	s_{32}
s_{03}	s_{13}	s_{23}	s_{33}

AES



AES(K,M) // $|M|$=128 and $|K|$=128

$(K_0,\ldots,K_{10})$<- KeySchedule(K) // $|K_i|$=128

$s \leftarrow M \oplus K_0$

for r=1 to 10

$s \leftarrow$ S(s)

$s \leftarrow$ ShiftRows(s)

if $r \leq 9$ then

$s \leftarrow$ MixCols(s)

$s \leftarrow s \oplus K_r$

Return s

AES – Byte Substitution layer (S-Box)

- An S-box has the following property
  - Identical i.e. same s-boxes per round
  - Nonlinear i.e. $S(s)+S(s') \neq S(s+s')$
  - Bijective i.e. $\exists_1$ 1-to-1 mapping of input and output bytes
    - S-box can be uniquely reversed
  - Implemented as a lookup table

s

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	C5

$\xrightarrow{S}$

s

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

AES – Diffusion Layer

- **Diffusion** over all input state bits
  - ShiftRows provides permutation of the data
  - Linear i.e. ShiftRows(s)+ShiftRows(s') = ShiftRows($s+s'$)
    - Similarly applies to *MixCols*

s

s_{00}	s_{10}	s_{20}	s_{30}
s_{01}	s_{11}	s_{21}	s_{31}
s_{02}	s_{12}	s_{22}	s_{32}
s_{03}	s_{13}	s_{23}	s_{33}

$\xrightarrow{\textit{ShiftRows}}$

s

s_{00}	s_{10}	s_{20}	s_{30}
s_{11}	s_{21}	s_{31}	s_{01}
s_{22}	s_{32}	s_{02}	s_{12}
s_{33}	s_{03}	s_{13}	s_{23}

AES – Diffusion Layer

- **Diffusion** over all input state bits
  - ShiftRows provides permutation of the data
  - Linear i.e. ShiftRows(s)+ShiftRows(s') = ShiftRows($s+s'$)
    - Similarly applies to *MixCols*
  - MixCols provides mix of blocks of 4 bytes

$$\overset{s}{\begin{bmatrix} s_{00} & s_{10} & s_{20} & s_{30} \\ s_{01} & s_{11} & s_{21} & s_{31} \\ s_{02} & s_{12} & s_{22} & s_{32} \\ s_{03} & s_{13} & s_{23} & s_{33} \end{bmatrix}} \xrightarrow{\textit{ShiftRows}} \overset{s}{\begin{bmatrix} \boxed{s_{00}} & s_{10} & s_{20} & s_{30} \\ \boxed{s_{11}} & s_{21} & s_{31} & s_{01} \\ \boxed{s_{22}} & s_{32} & s_{02} & s_{12} \\ \boxed{s_{33}} & s_{03} & s_{13} & s_{23} \end{bmatrix}} \cdot \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \xrightarrow{\textit{MixCols}} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}$$

AES – Key Addition Layer

$$\overset{s}{\begin{bmatrix} s_{00} & s_{10} & s_{20} & s_{30} \\ s_{01} & s_{11} & s_{21} & s_{31} \\ s_{02} & s_{12} & s_{22} & s_{32} \\ s_{03} & s_{13} & s_{23} & s_{33} \end{bmatrix}} \oplus \overset{K_r}{\begin{bmatrix} k_{00} & k_{10} & k_{20} & k_{30} \\ k_{11} & k_{21} & k_{31} & k_{01} \\ k_{22} & k_{32} & k_{02} & k_{12} \\ k_{33} & k_{03} & k_{13} & k_{23} \end{bmatrix}} \longrightarrow \overset{s}{\begin{bmatrix} s_{00} & s_{10} & s_{20} & s_{30} \\ s_{01} & s_{11} & s_{21} & s_{31} \\ s_{02} & s_{12} & s_{22} & s_{32} \\ s_{03} & s_{13} & s_{23} & s_{33} \end{bmatrix}}$$

AES

- S-Boxes provide confusion
- ShiftRows and MixCols provide diffusion
- Key Addition Layer protects against inverting attacks

AES is considered secure because

- *conjectured* to be a good **pseudorandom permutation** function
- got no *serious* **cryptoanalysis** attacks so far

AES Question

Select the **correct** option:

a) AES uses keys with 64 bits.

b) AES has no known practical attacks against it.

c) AES is mathematically proven to be secure if factoring large prime integers is hard.

d) AES can use keys of any length.

Hash and MAC

integrity

Source: Bansky

Hash functions

Common building block of security mechanisms

- compare by hash
- virus protection
- OTP
- storing passwords
- fundamental ingredient for many crypto primitives

Hash functions

Definition (Hash function)

- A function $\mathcal{H}$ that takes an arbitrary block of data and returns a fixed-size bit string (digest)

E.g. $\mathcal{H}$('fox') = b99c21513df8309c021977902526e2f3881758a1

E.g. $\mathcal{H}$('The red fox jumps over the blue dog') = 0504e140d01c8c8cad73ac18873fd7944e236f90

E.g. $\mathcal{H}$('The red fox **b**umps over the blue dog')= 78e883a20497df7af2ba0d4dff062a26137c024d

E.g. $\mathcal{H}$('The red fox jumps over the blue dog**s**')= 8ee7cb3ea20307bbb68bee60fd1c3068aa28b455

- Goal: integrity. How?

Cryptographic Hash functions

Cryptographic hash function requirements

- Pre-image resistance (one-way): Given h=$\mathcal{H}(m)$ is infeasible to find m

- Second pre-image resistance: Given m_1 is infeasible to find $m_2 \neq m_1 . \mathcal{H}(m_1) = \mathcal{H}(m_2)$
  - Second pre-image implies pre-image resistance (why?)

- Collision resistance: It is infeasible to find $\mathcal{H}(m_1) = \mathcal{H}(m_2)$ and $m_1 \neq m_2$
  - Collision resistance implies second pre-image resistance (why?)

Industry Sandards: SHA2 (being deprecated), SHA3

Hash Functions Question

Select the **correct** option:

a) Hash functions protect message confidentiality.

b) Hash function output length depends on the input length.

c) Given a hash function output its hard to find the input that yields that output.

d) It is easy to find two inputs of a hash function that yield the same output.

Hash functions: Application

Storing passwords

UserID	password
brun	qwerty949
rosg	incorrect
rikj	asdfg
maca	944aaa
debois	asdfg

- What if db is compromised?

Hash functions: Application

Storing passwords

UserID	$\mathcal{H}$(password)
brun	1977902526e2f3881758a1
rosg	73ac18873fd7944e236f90
rikj	ba0d4dff062a26137c024d
maca	68bee60fd1c3068aa28b45
debois	ba0d4dff062a26137c024d

- Is it fixed now?

Hash functions: Application

Storing passwords

UserID	$\mathcal{H}$(password)
brun	1977902526e2f3881758a1
rosg	73ac18873fd7944e236f90
rikj	ba0d4dff062a26137c024d
maca	68bee60fd1c3068aa28b45
debois	ba0d4dff062a26137c024d

- It is possible to identify users who have the same password.

Hash functions: Application

Storing passwords - Salting

UserID	Salt	$\mathcal{H}$(password|salt)
brun	4738295	3881758a11977902526e2f
rosg	3727283	jej48929dj38d833838ddj39
rikj	3838759	dkkeoe33392lj39d84939dk
maca	9048040	4849dj29d9ke93304kf94k4
debois	2872900	48d83jj9d2kk334449dk9s9

- A random string called a ``salt'' is concatenated with the password.

Message Authentication Codes (MAC)

- Goal: integrity + authenticity
  - No confidentiality!

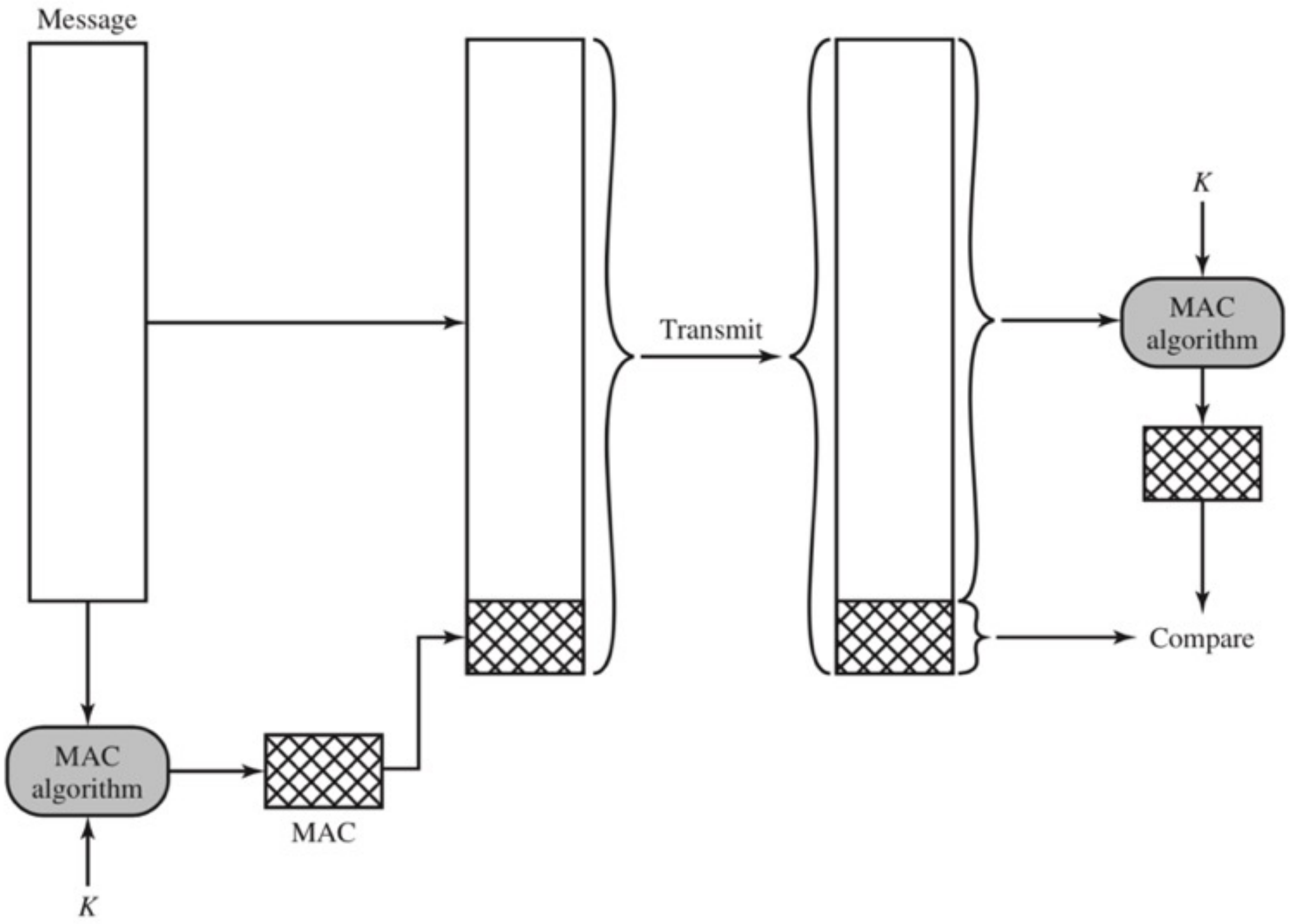
Message Authentication Codes (MAC)

- Goal: integrity + authenticity
  - No confidentiality!

- An algorithm tag = mac(m,k)
- An algorithm d = ver(tag,m,k)
  - k = KeyGen() being a random string

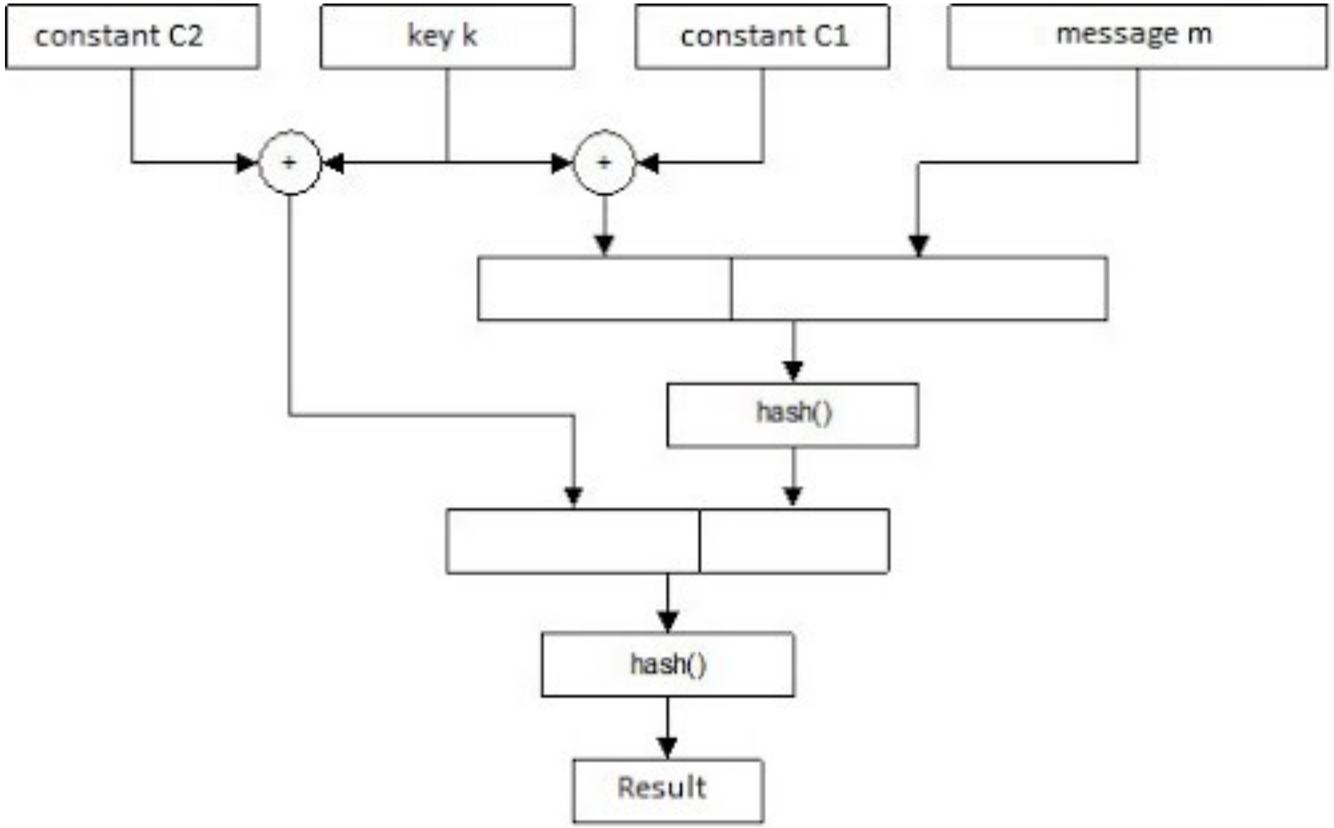
- Correctness: ver(mac(m,k),m,k) = `true`

HMAC



Message
K
MAC
algorithm
Transmit
MAC
algorithm
MAC
K
Compare

HMAC

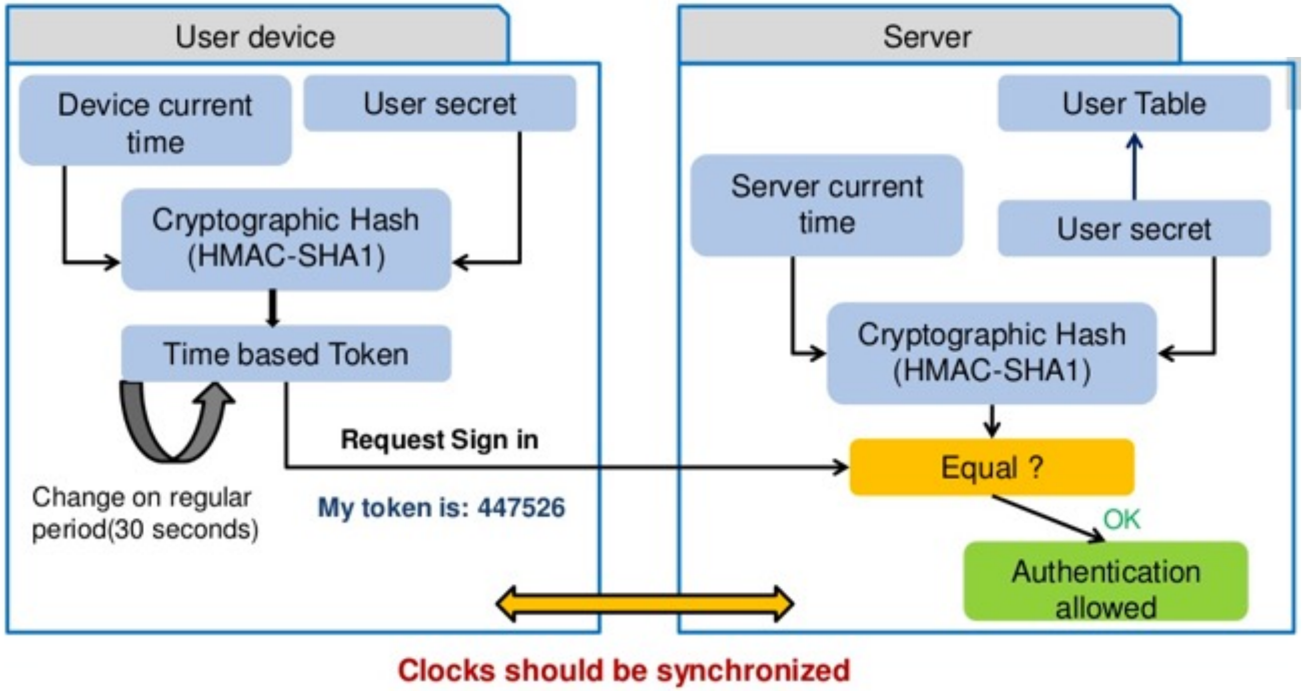


MAC = $\mathcal{H}((k \oplus const1) \mid \mathcal{H}(k \oplus const2 \mid m))$

- const1 and const2 constants and public

MAC Application

Smart Token



Hash Functions Question

Select the **incorrect** option:

a) MACs protect message integrity.

b) MACs do not require keys.

c) MACs have constant output length.

d) HMAC can be constructed from any cryptographically strong hash function (e.g. SHA3).

Limitations of symmetric cryptography

- Sender and Receiver should meet in person and choose k

- Need a key for each pair of agents who want to communicate securely

- How to share a secret key securely between two agents over an insecure network?

Summary

- Symmetric Cryptography
  - Definition: correctness and security
  - AES: 3 layers
  - Limitations: how to share a key?
- Hashing and MAC
  - Definition: collision resistance
  - HMAC: goals
  - Applications: TOTP, password storing
- Next lecture: key exchange and asymmetric cryptography.