IT-Security (ITS) B1

DIKU, E2020

Lecture plan

36	31 Aug	10-12	TL	Introduction, security concepts and the threat of hacking
	04 Sep	10-12	TL	Buffer overflow
37	07 Sep	10-12	CJ	Software security, Operating system security
	11 Sep	10-12	CJ	User authentication and access control
38	14 Sep	10-12	TL	Malicious software
	18 Sep	10-12	CJ	Firewalls and denial-of-service attacks
39	21 Sep	10-12	CJ	Cloud and IoT
	25 Sep	10-12	TL	Cryptography
40	**28 Sep**	**10-12**	**TL**	**Internet security protocols**
	02 Oct	10-12	TL	Intrusion detection
41	05 Oct	10-12	TL	Forensics
	09 Oct	10-12	CJ	IT security management
42				Fall Vacation - No lectures
43	19 Oct	10-12	CJ	Privacy 1
	23 Oct	10-12	CJ	Privacy 2
44	26 Oct	10-11	Guest	Final guest lecture
		11-12	All	Recap and Q/A
45	xx Nov			Exam

Lecture plan

Mandag d. 28. september
- kl. 10-12 Cryptography

Fredag d. 2. oktober
- kl. 09-10 Internet security protocols (bemærk ekstra time fra kl. 9 allerede)
- kl. 10-12 Intrusion detection

Mandag d. 5. oktober
- kl. 9-11 Forensics (bemærk flyttet fra kl. 10-12 til kl. 09-11)

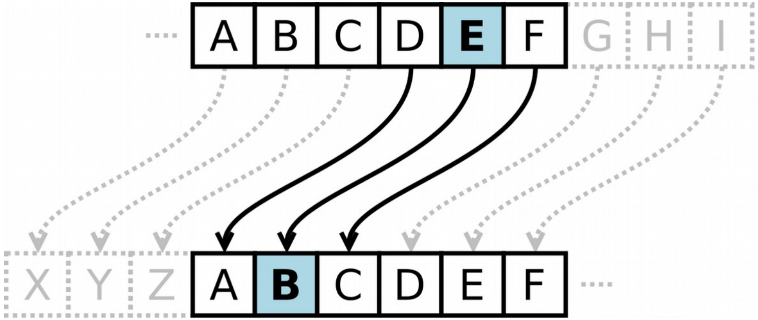
Today’s agenda

Part 1: Crypto building blocks

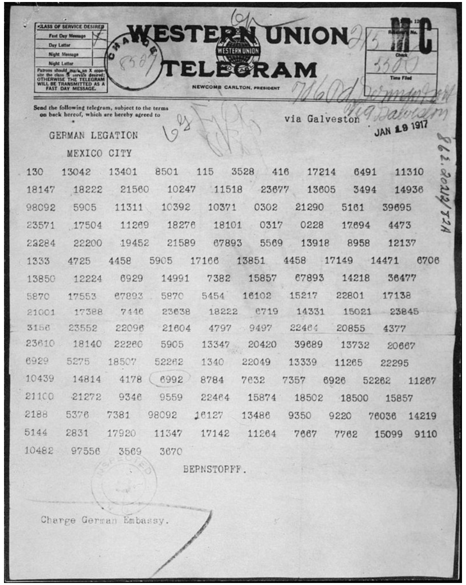
Part 2: More crypto building blocks

(Next time: Real-world crypto protocols)

Cryptography influence world events



Cryptography influence world events


CLASS OF SERVICE DESIRED
Fast Day Message
Day Letter
Night Message
Night Letter
Patrons should mark an X opposite the class of service desired; OTHERWISE THE TELEGRAM WILL BE TRANSMITTED AS A FAST DAY MESSAGE.

WESTERN UNION TELEGRAM

NEWCOMB CARLTON, PRESIDENT

Receiver's No.
Check
Time Filed

Send the following telegram, subject to the terms on back hereof, which are hereby agreed to

via Galveston

JAN 19 1917

GERMAN LEGATION

MEXICO CITY

130 13042 13401 8501 115 3528 416 17214 6491 11310
18147 18222 21560 10247 11518 23677 13605 3494 14936
98092 5905 11311 10392 10371 0302 21290 5161 39695
23571 17504 11269 18276 18101 0317 0228 17694 4473
22284 22200 19452 21589 67893 5569 13918 8958 12137
1333 4725 4458 5905 17166 13851 4458 17149 14471 6706
13850 12224 6929 14991 7382 15857 67893 14218 36477
5870 17553 67893 5870 5454 16102 15217 22801 17138
21001 17388 7446 23638 18222 6719 14331 15021 23845
3156 23552 22096 21604 4797 9497 22464 20855 4377
23610 18140 22260 5905 13347 20420 39689 13732 20667
6929 5275 18507 52262 1340 22049 13339 11265 22295
10439 14814 4178 6992 8784 7632 7357 6926 52262 11267
21100 21272 9346 9559 22464 15874 18502 18500 15857
2188 5376 7381 98092 16127 13486 9350 9220 76036 14219
5144 2831 17920 11347 17142 11264 7667 7762 15099 9110
10482 97556 3569 3670

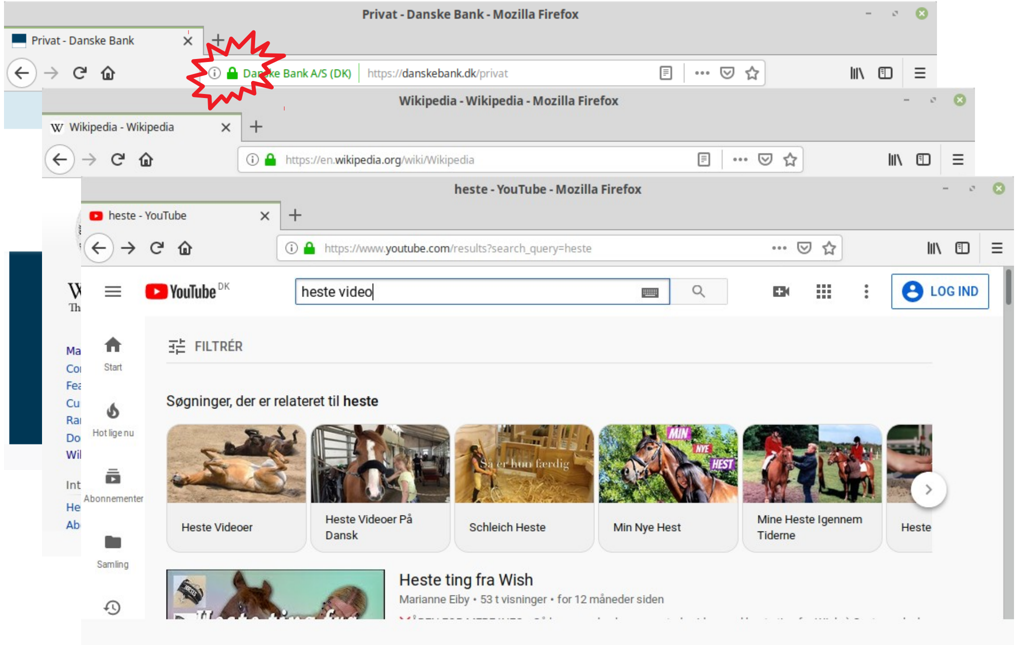
BERNSTORFF.

Charge German Embassy.

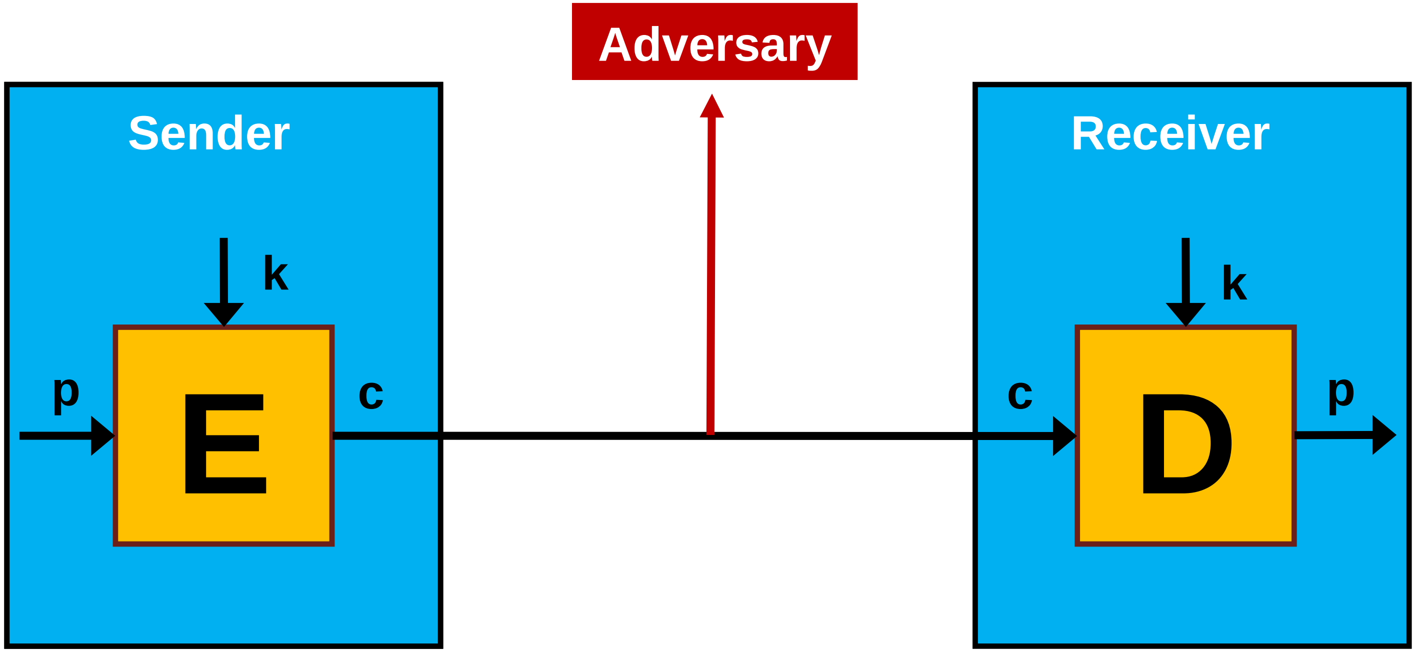
EXPLODING IN HIS HANDS.
—Kirby in the New York *World*.

Cryptography influence world events

Our goal: Secure communication



Cryptosystems



Security goals

Confidentiality

Integrity

Authenticity

Non-repudiation

Warm-up question

FileCrypt

“**FileCrypt** is a dynamic non-factor based quantum AI encryption hardware solution.

Developed by our cryptographic experts and hardwired into a tamper-resistant USB token.

Plug the token into your PC, start the program and encrypt the files you need to protect”

What problems do you see with this solution?

Multiple concerns

#1: “Developed by our cryptographic experts”

Should we trust proprietary crypto over public peer-reviewed time-tested crypto?

#2: “Dynamic non-factor based quantum AI”

What does that mean? Are there any academic papers that discuss this concept?

#3: “Plug the token into your PC”

Can anyone do this? What if token is lost? Violates **Kerckhoffs’ Principle**

Kerckhoffs' (2nd) Principle

"Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi

The security of a cryptographic algorithm must rest solely in the secrecy of its **key**, not in the secrecy of the algorithm itself

Collaries:
Assume attacker knows the algorithm
Make it available for public analysis
Protect the key!

Auguste Kerckhoffs (1835 – 1903)

Type of Attack	Known to Cryptanalyst
Ciphertext only	•Encryption algorithm •Ciphertext to be decoded
Known plaintext	•Encryption algorithm •Ciphertext to be decoded •One or more plaintext-ciphertext pairs formed with the secret key
Chosen plaintext	•Encryption algorithm •Ciphertext to be decoded •Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen ciphertext	•Encryption algorithm •Ciphertext to be decoded •Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen text	•Encryption algorithm •Ciphertext to be decoded •Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key •Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Symmetric cryptosystems

Symmetric cryptosystems

Stream ciphers

One time pad

If *k* random, |*k*| >= |*p*|, never reused, and kept secret, then then impossible to decrypt or break without knowing the key (Shannon, 1949)

Key	0	1	0	1	1	1	0	0	1	0	
Plaintext	1	1	0	0	0	1	1	0	0	0	$\oplus$
Ciphertext	1	0	0	1	1	0	1	0	1	0	

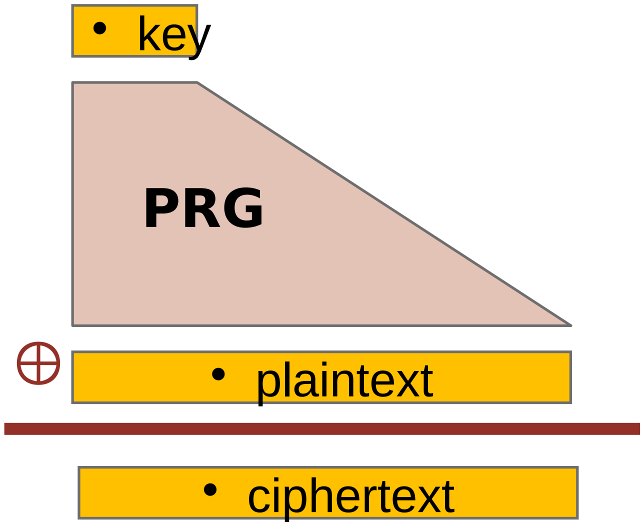
Towards modern stream ciphers

Problem

- OTP key as long as plaintext

Solution

- Generate pseudo random keystream



1st rule of stream ciphers

Never reuse key

$$C_1 \leftarrow P_1 \oplus PRG(k)$$

$$C_2 \leftarrow P_2 \oplus PRG(k)$$

$$C_1 \oplus C_2 \rightarrow P_1 \oplus P_2$$

$$P_1 \oplus P_2 \rightarrow P_1, P_2$$

Solution: Initialisation Vector (IV)

For each message

- Generate IV
- Mix k with IV
- Generate keystream PRG(k+IV) and encrypt
- Send c and IV (in plaintext)

Change k before IVs run out

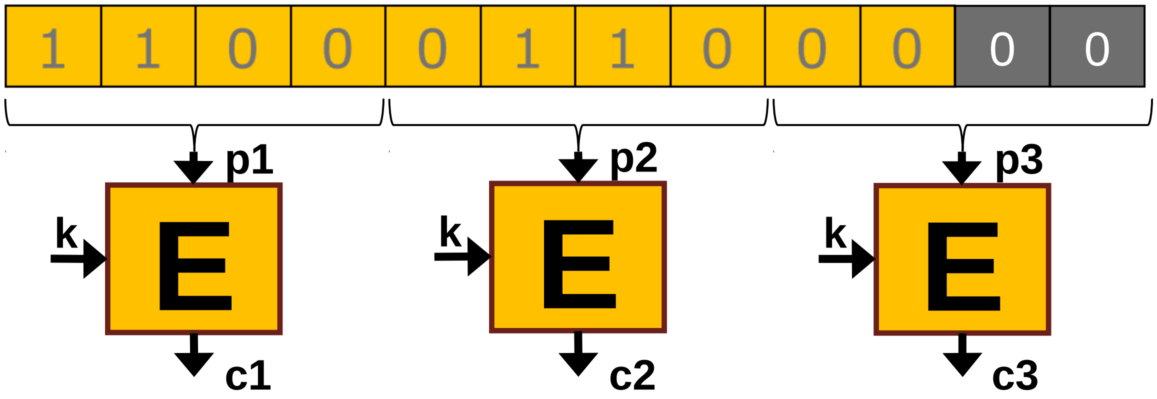
Stream ciphers in the wild


Block ciphers

Block ciphers

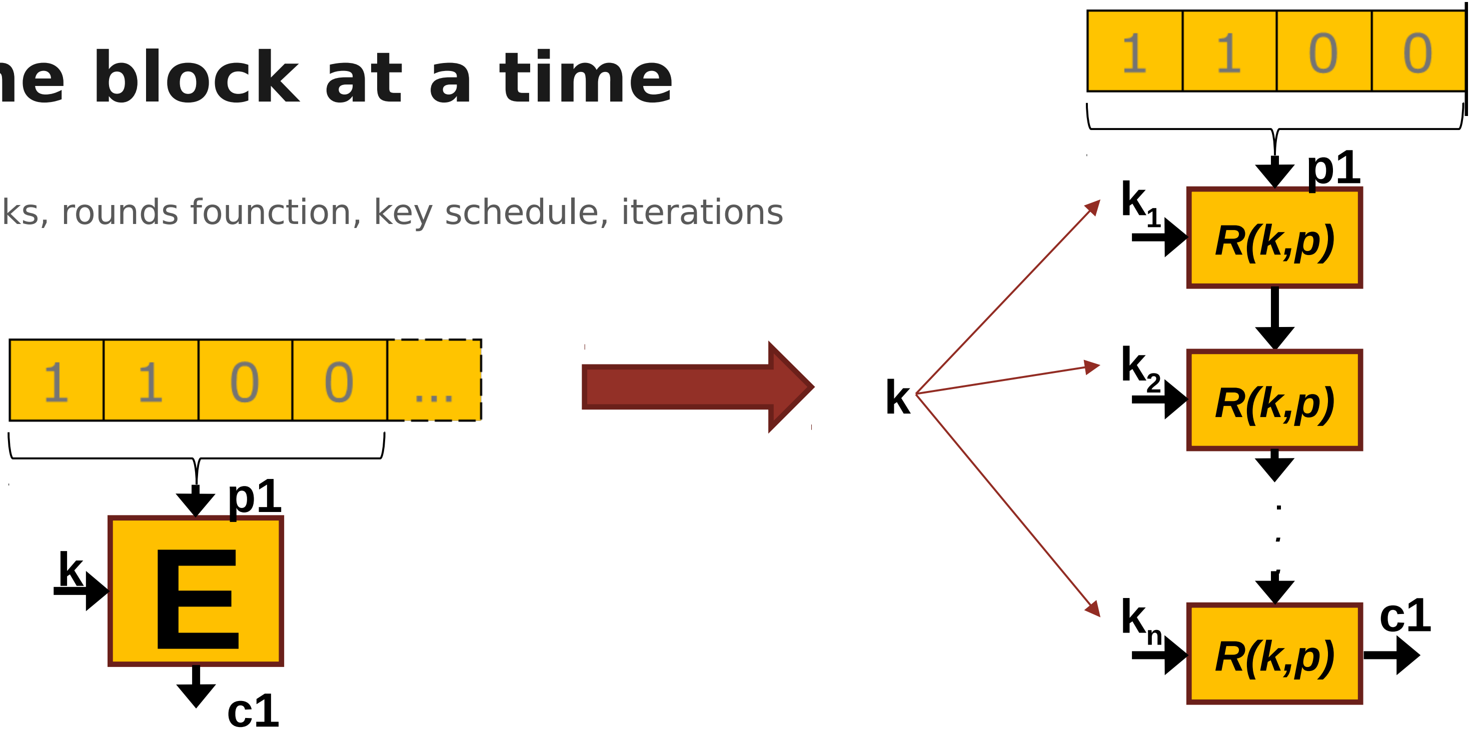
One block at a time – as oppossed to one bit at a time

- **padding**

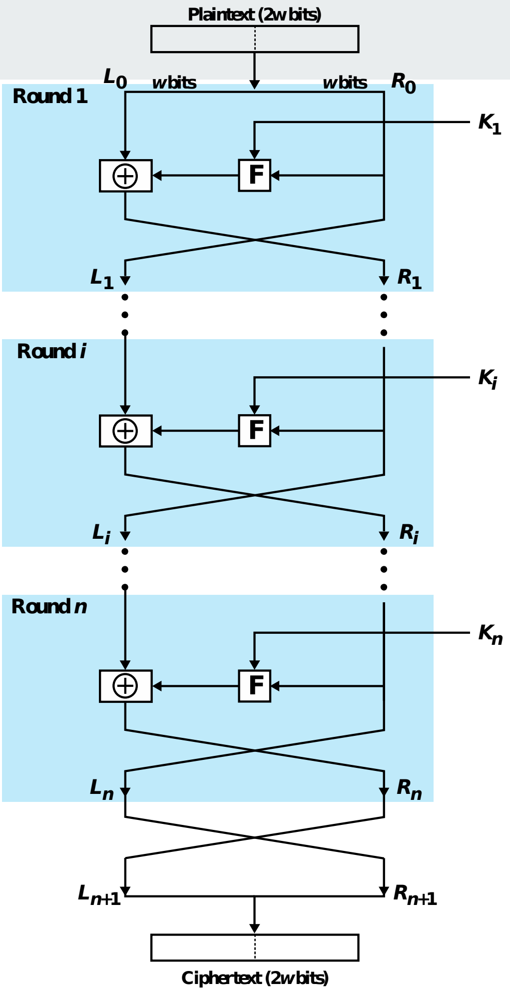


One block at a time

Blocks, rounds founction, key schedule, iterations



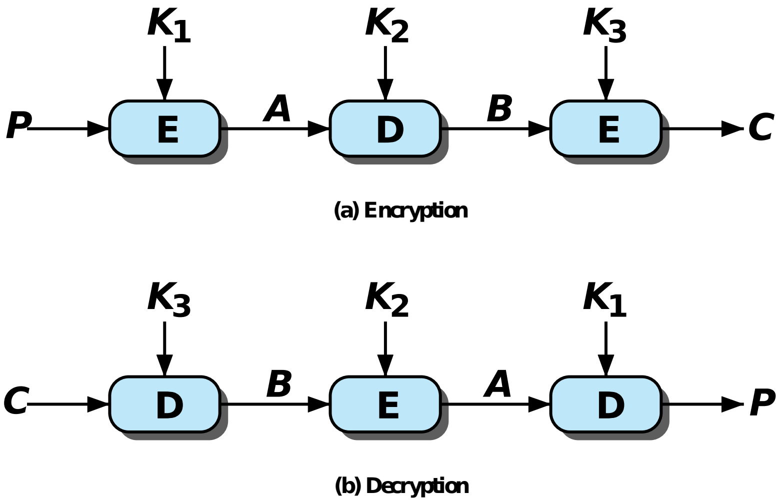
Feistel network



DES

DES

Key 64, block 64, rounds 16



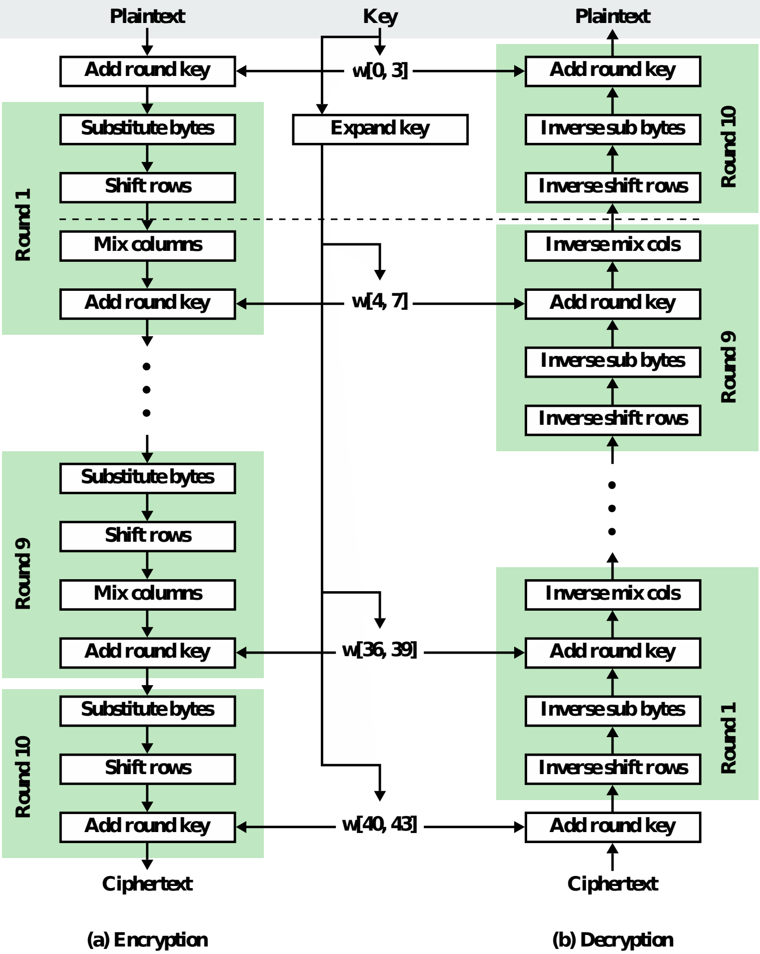
AES

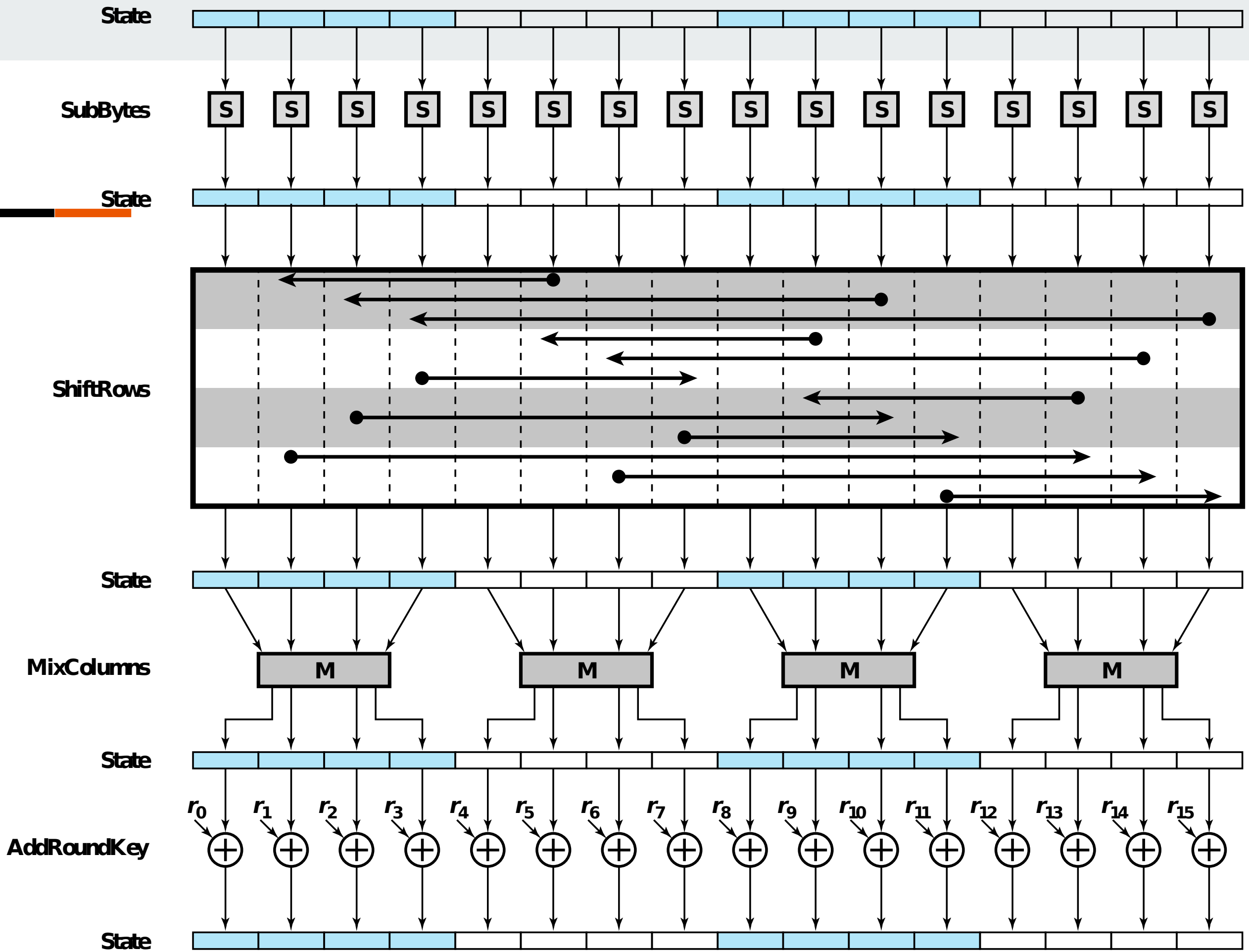
AES

Keys 128/192/256

Block 128

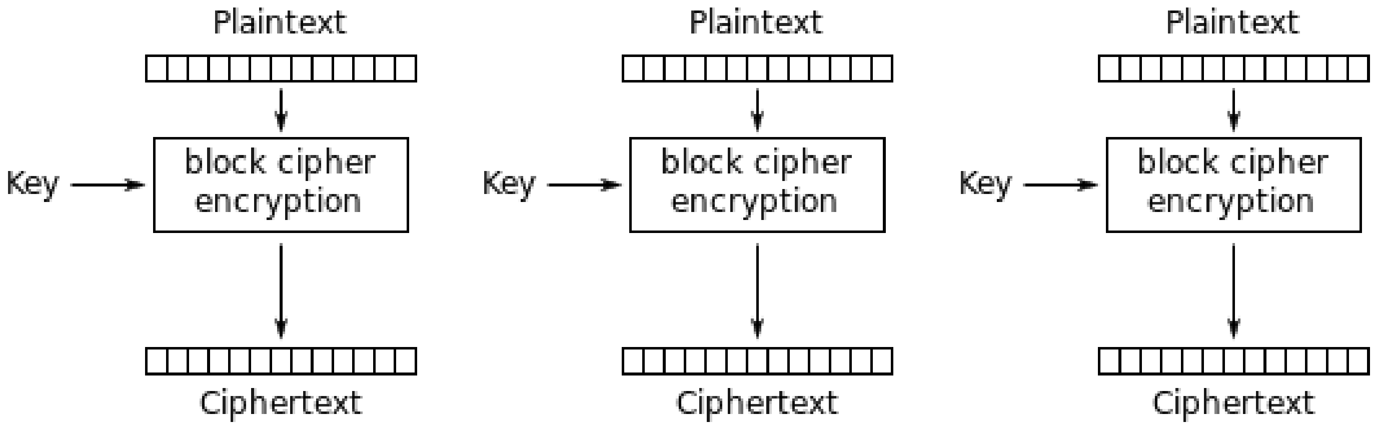
Rounds 10/12/14




State
SubBytes
S
State
ShiftRows
State
MixColumns
M
State
AddRoundKey
r_0 r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8 r_9 r_{10} r_{11} r_{12} r_{13} r_{14} r_{15}
State

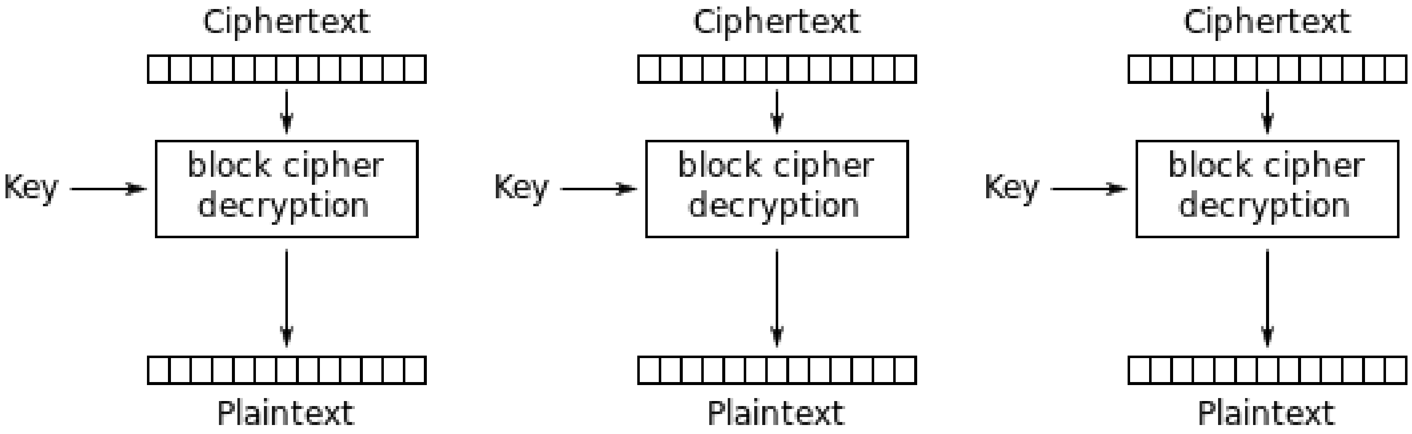
Modes of operation

Electronic Codebook (ECB)



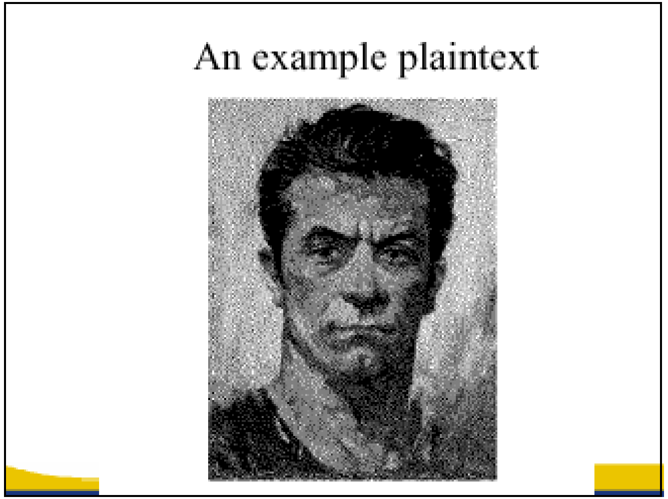
Electronic Codebook (ECB) mode encryption

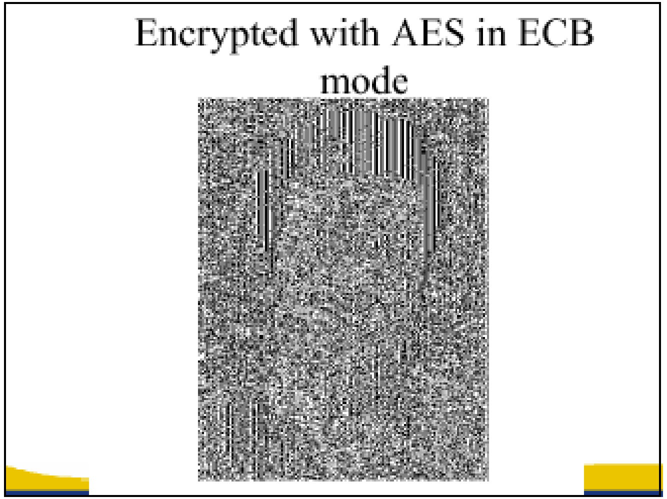
ECB decyption



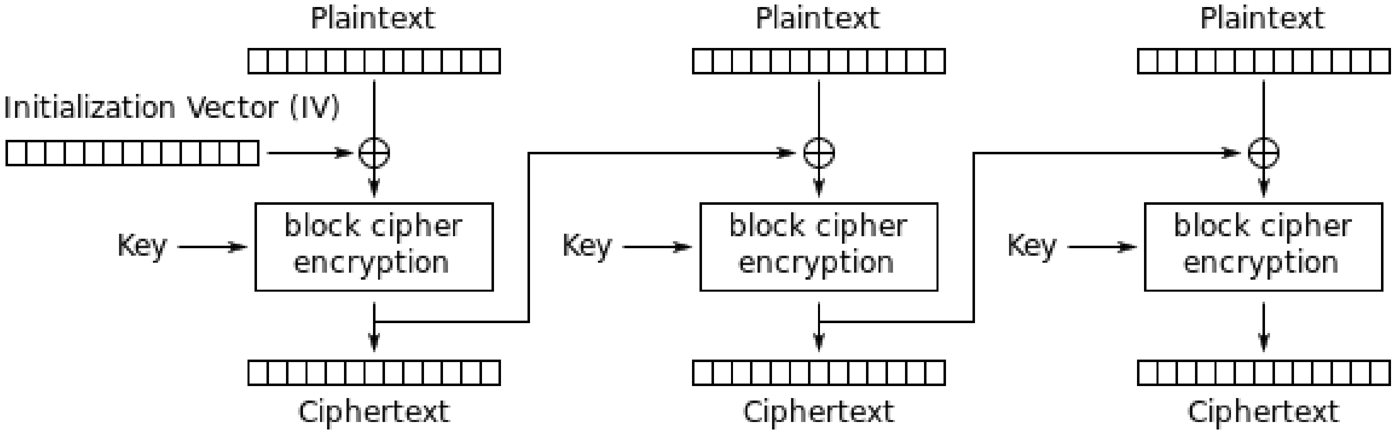
Electronic Codebook (ECB) mode decryption

If p1 = p2, then c1 = c2



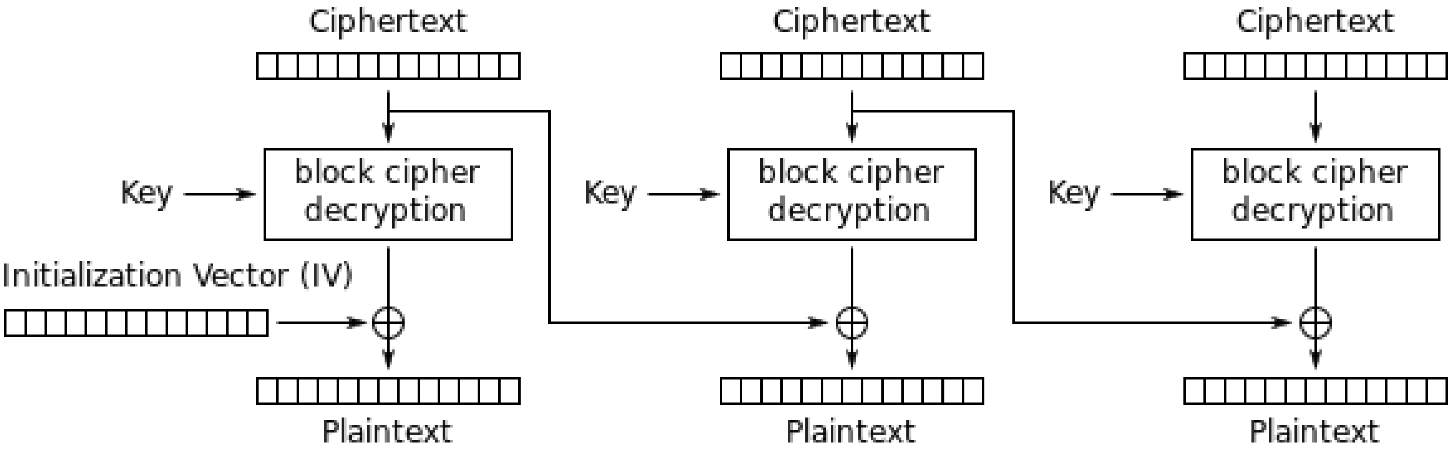


Cipher Block Chaining



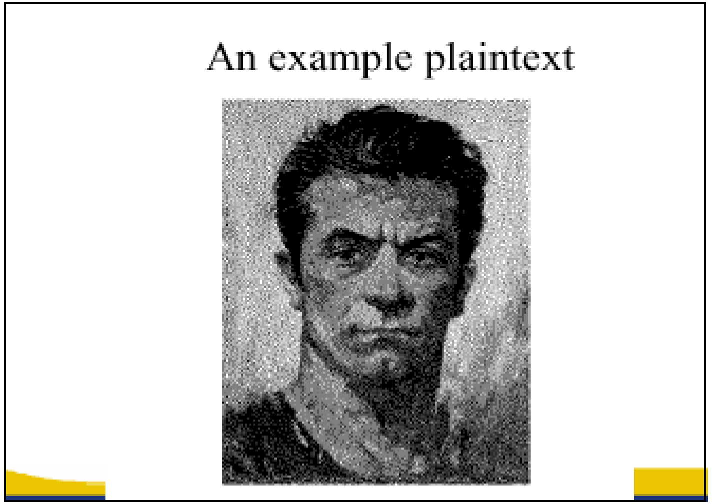
Cipher Block Chaining (CBC) mode encryption

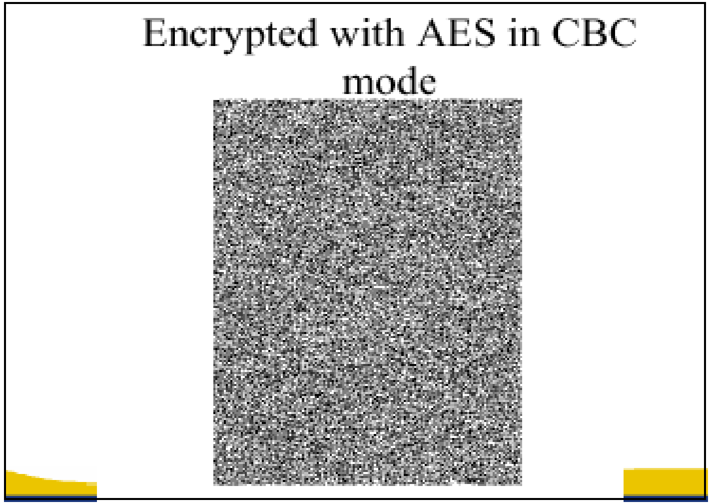
CBC decryption



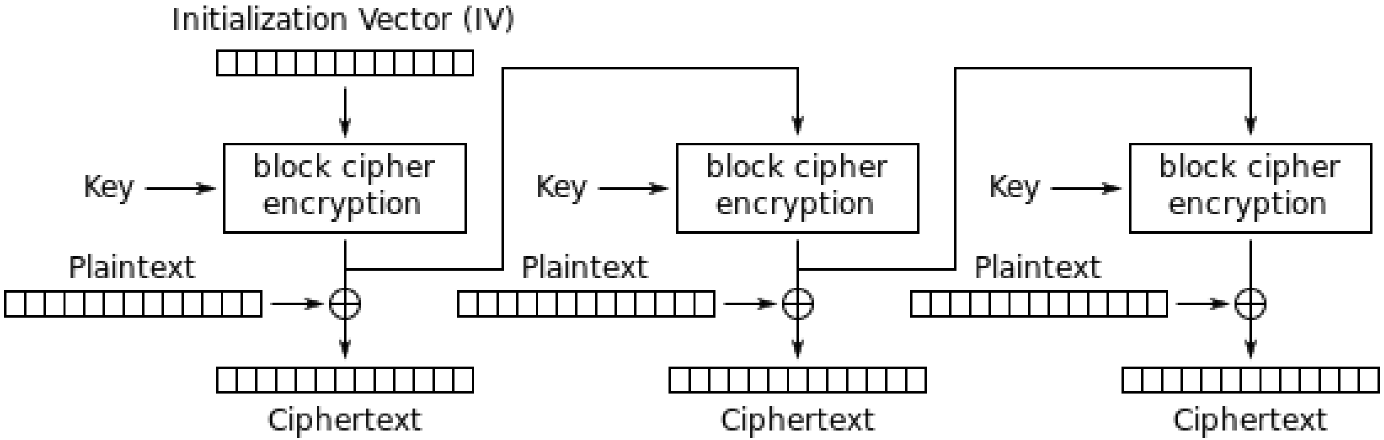
Cipher Block Chaining (CBC) mode decryption

Better





Output Feedback



Output Feedback (OFB) mode encryption

Security goals revisited

“Susceptibility to malicious insertions and modifications. Because each symbol is separately enciphered, an active interceptor who has broken the code can splice together pieces of previous messages and transmit a spurious new message that may look authentic.” - Phleeger & Phleeger in Security in Computing, Pearson, 2003

Is this a disadvantage of stream cipher? Why, why not?

Security goal of encryption: Confidentiality

Status

Confidentiality: Check!

Integrity: Missing

Message authentication code (MAC)

Message authentication code

Goal: Provide integrity

Process

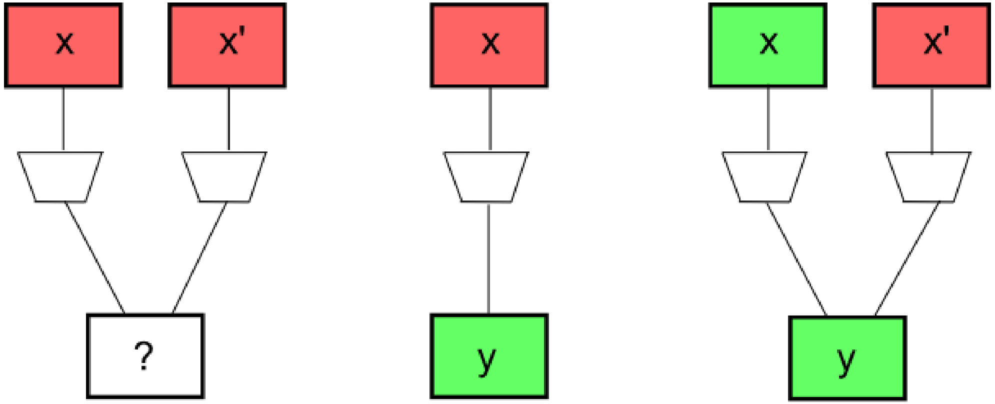
Choose a cryptographic hash funciton h : $\{0,1\}^x \rightarrow \{0,1\}^n$

Sender: Send h(m),m

Receiver: Calculate h(m) and verify it matches h(m)

Examples MD5 (n = 128), SHA-256 (n = 256)

Cryptographic hash functions



Finding Collision

Finding Inversion

Finding 2nd Pre-image

Hash-based MAC (HMAC)

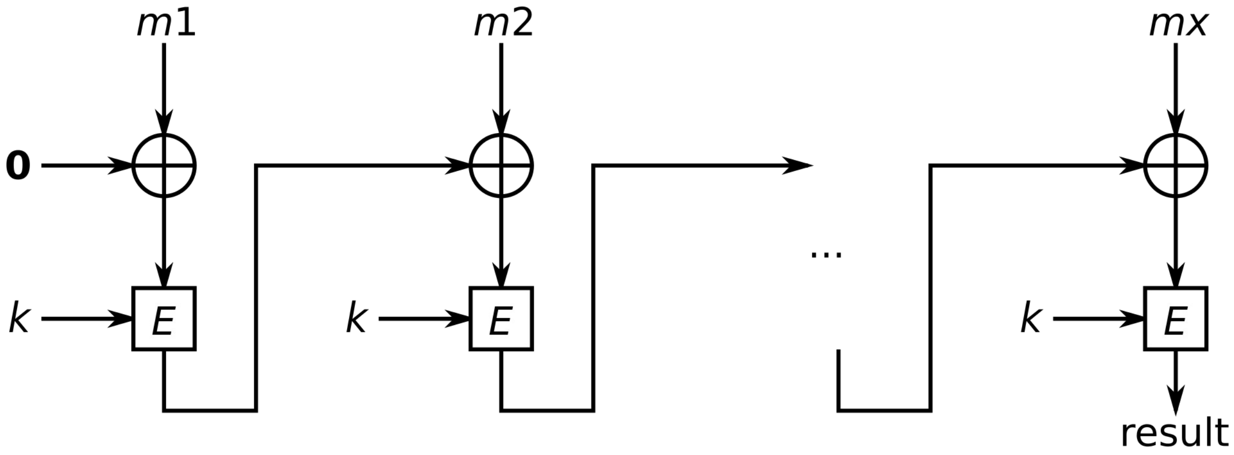
RFC2104: Hash-based MAC

HMAC(h,k,m) =

$$h\left((k \oplus opad) \parallel h((k \oplus ipad) \parallel m)\right)$$

HMAC provides integrity and authenticity

CBC-MAC



Car keys

Your car key sends the code for ”open the door”, together with a MAC, to the car whenever you press the button.

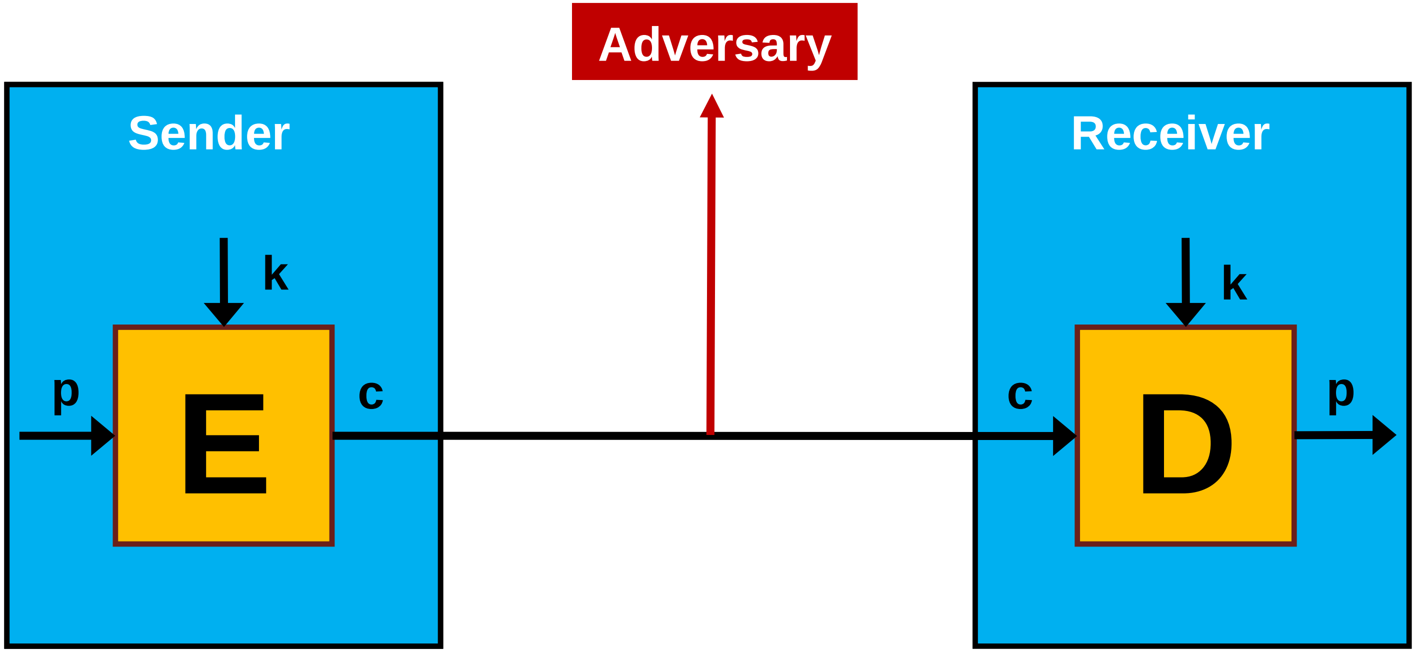
What could go wrong?

Replay attack: attacker records message and replays it later

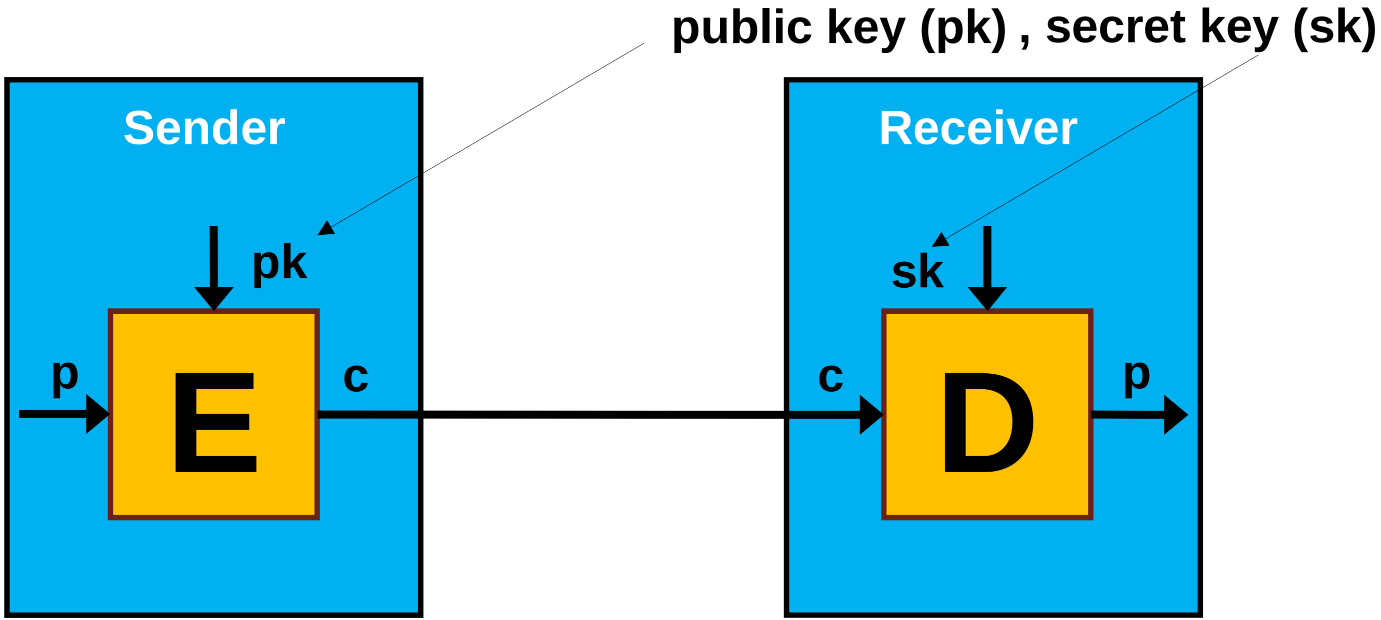
We need some freshness: a timestamp or nonce

Non-repudiation

Cryptosystems



Enter: Asymmetric encryption



Analogy: Combination locks

Bob sends out locks with combination he only knows

Alice picks one of Bob's locks, places her message in a box and locks it with Bob's lock

Bob is the only one who can open the box now

No pre-shared key!

Bob

- Publish public key, protect private key

Alice

- Encrypt message with Bob’s public key

Bob

- Decrypts with his private key

Rivest Shamir Adleman (RSA), 1978

First asymmetric cryptosystem

RSA encryption and decryption

Public key (N,e), private key (d)

$C = M^e \pmod{N}$

$M = C^d \pmod{N}$

Asymmetric encryption: Yes! But what about non-repudiaton?

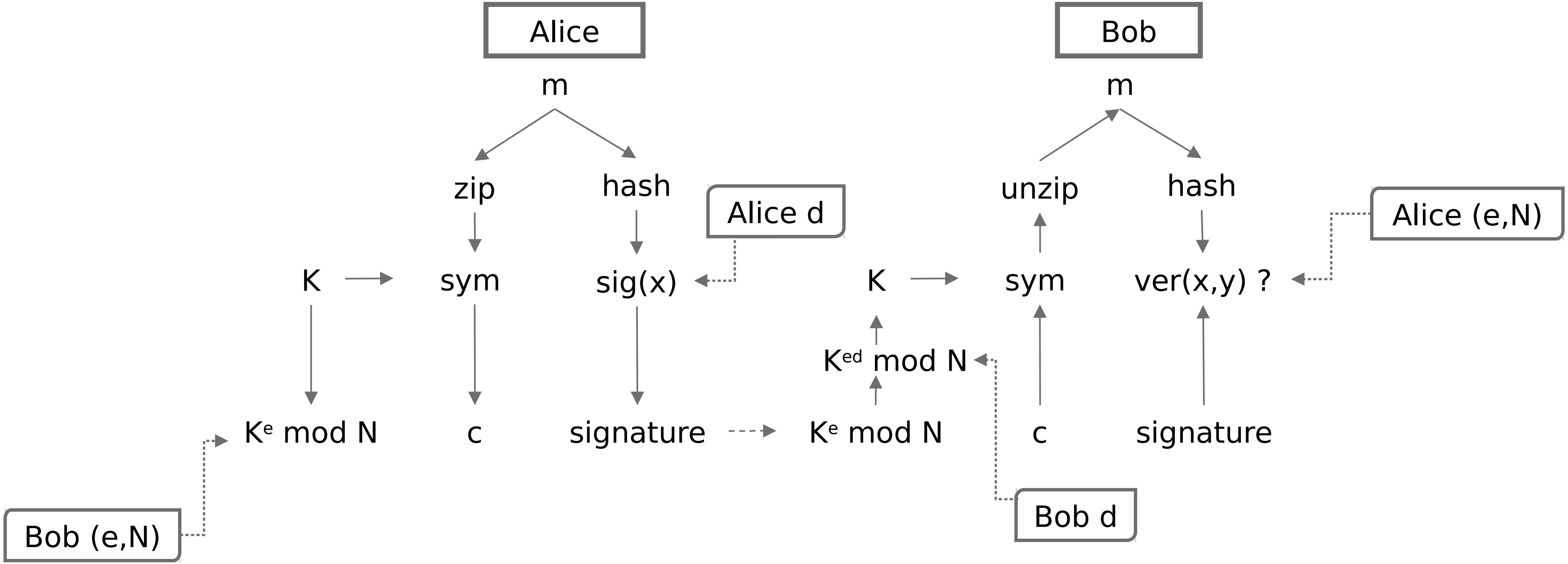
Reverse

Public key (N,e), private key (d)

Signature $sig(M) = M^d \pmod N$

Verify $ver(M,sig(M)) = \text{true}$ iff $M = (M^d)^e \pmod N$

Putting it all togehter



Next time, real-world crypto protocols

Wrap-up

Security goals achieved

Confidentiality

Integrity

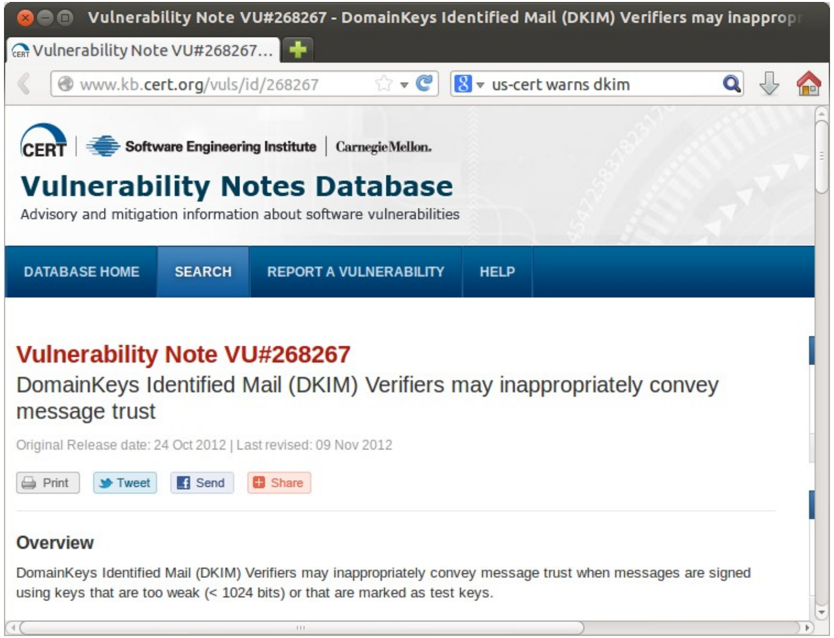
Authenticity

Non-repudiation

CHECK!

But crypto can still fail

Small keys fail



Collision fail


ars technica

See what Accuweather built for Windows

MAIN MENU MY STORIES: 25 FORUMS SUBSCRIBE VIDEO

RISK ASSESSMENT / SECURITY & HACKTIVISM

Crypto breakthrough shows Flame was designed by world-class scientists

The spy malware achieved an attack unlike any cryptographers have seen before.

by Dan Goodin - June 7 2012, 8:20pm -200

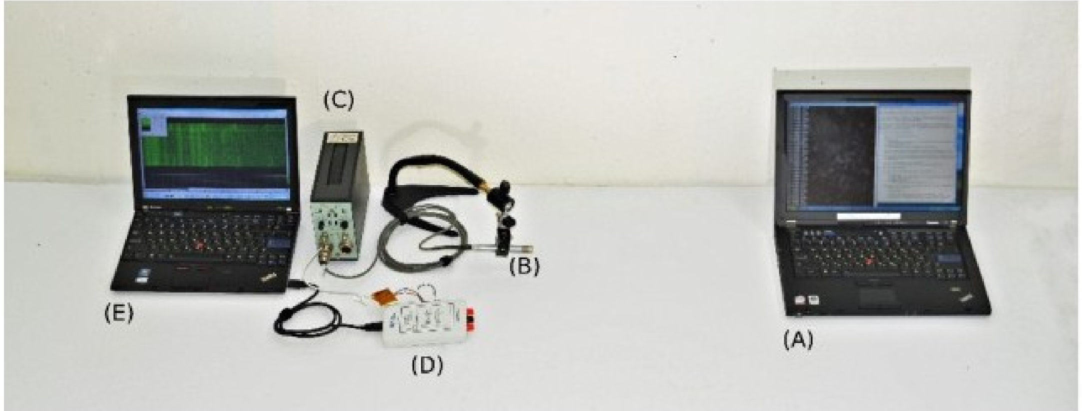
BLACK HAT NATIONAL SECURITY 161

Impressive fail

New attack steals e-mail decryption keys by capturing computer sounds

Scientists use smartphone to extract secret key of nearby PC running PGP app.

by **Dan Goodin** - Dec 18, 2013 11:25 pm UTC



Bad choice fail

IRS Encourages Poor Cryptography

Buried in one of the documents are the rules for encryption:

> While performing AES encryption, there are several settings and options depending on the tool used to perform encryption. IRS recommended settings should be used to maintain compatibility:
>
> - Cipher Mode: ECB (Electronic Code Book).
> - Salt: No salt value
> - Initialization Vector: No Initialization Vector (IV). If an IV is present, set to all zeros to avoid affecting the encryption.
> - Key Size: 256 bits / 32 bytes Key size should be verified and moving the key across operating systems can affect the key size.
> - Encoding: There can be no special encoding. The file will contain only the raw encrypted bytes.
> - Padding: PKCS#7 or PKCS#5.

ECB? Are they serious?

DIY fail

Smart grid security WORSE than we thought

OSGP's DIY MAC is a JOKE

11 May 2015 at 02:03, Richard Chirgwin 222 36 22

Backdoor fail

Topic: Security *Follow via:*

NIST finally dumps NSA-tainted random number algorithm

Summary: *Many years since a backdoor was discovered, probably planted by the NSA, public pressure finally forces NIST to formally remove Dual_EC_DRBG from their recommendations.*

By Larry Seltzer for Zero Day | April 23, 2014 -- 14:04 GMT (07:04 PDT)

Follow @lseltzer

Comments 2 Vote 1 more +

Supply chain fail

Schneier on Security

Blog Newsletter Books Essays News Talks Academic About Me

Home > Blog

Crypto AG Was Owned by the CIA

The Swiss cryptography firm Crypto AG sold equipment to governments and militaries around the world for decades after World War II. They were owned by the CIA:

> But what none of its customers ever knew was that Crypto AG was secretly owned by the CIA in a highly classified partnership with West German intelligence. These spy agencies rigged the company's devices so they could easily break the codes that countries used to send encrypted messages.

This isn't really news. We have long known that Crypto AG was backdooring crypto equipment for the Americans. What is new is the formerly classified documents describing the details:

> The decades-long arrangement, among the most closely guarded secrets of the Cold War, is laid bare in a classified, comprehensive CIA history of the operation obtained by The Washington Post and ZDF, a German public broadcaster, in a joint reporting project.
>
> The account identifies the CIA officers who ran the program and the

Search

Powered by *DuckDuckGo*

Go

Blog Essays Whole site

Subscribe

About Bruce Schneier

Malware fail

Krypto-guru: Tiden er ved at løbe fra kryptering | Version2 - Mozilla Firefox (Private B

IT-NYHEDER BLOGS IT-JOB IT-FIRMAER WHITEPAPERS

EMNER *Cyberkrig, It-sikkerhed, Kryptering, Malware-virus* Se kommentarer (4)

Krypto-guru: Tiden er ved at løbe fra kryptering

Adi Shamir, en af opfinderne af RSA-kryptering, mener, at kryptografiens rolle til dels bliver udspillet af nye typer avanceret malware, som ofte stammer fra regeringer.

Af ***Mikkel Meister*** *Tirsdag, 5. marts 2013 - 16:16*

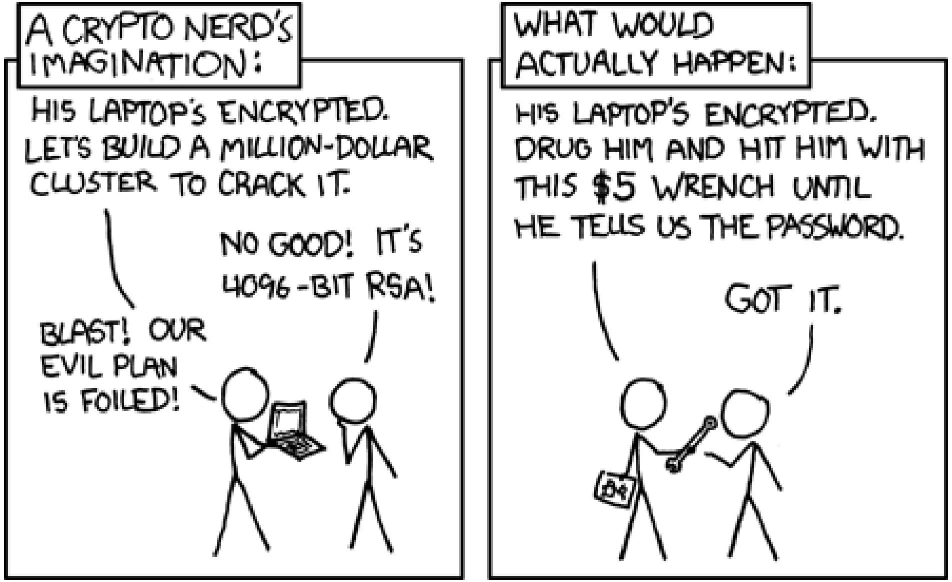
Kryptografi har ikke længere den samme vigtige rolle for it-sikkerhed som tidligere.

Det siger en af feltets helt store personligheder, Adi Shamir, som er en af skaberne af public key-kryptering, ifølge flere udenlandske it-medier.

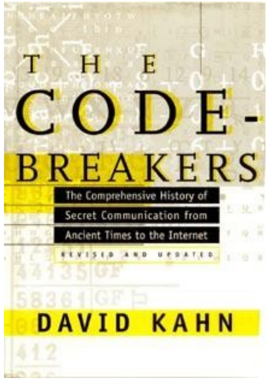
Han henviser til højtprofilerede angreb de senere år, hvor selv de mest hærdede it-sikkerhedsstrukturer omgås af statssponsorerede hackere og avanceret malware som Stuxnet.

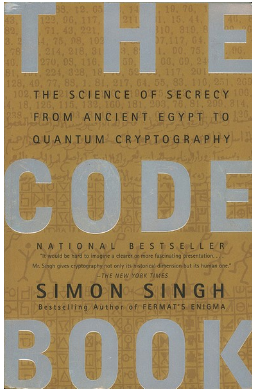
»Jeg mener afgjort, at kryptografi er ved at blive mindre afgørende. Selv de mest sikre it-systemer på de mest isolerede steder er blevet gennemhullet i de senere år af flere APT'er (Advanced Persistant Threat eller statsmalware, *red.*) og andre avancerede typer af angreb,« sagde Adi Shamir på RSA 2013-konferencen i sidste uge, skriver Threatpost.com.

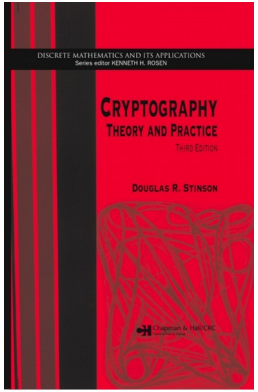
Real-world fail



Suggested reading







Lecture plan

Mandag d. 28. september
- kl. 10-12 Cryptography

Fredag d. 2. oktober
- kl. 09-10 Internet security protocols (bemærk ekstra time fra kl. 9 allerede)
- kl. 10-12 Intrusion detection

Mandag d. 5. oktober
- kl. 9-11 Forensics (bemærk flyttet fra kl. 10-12 til kl. 09-11)