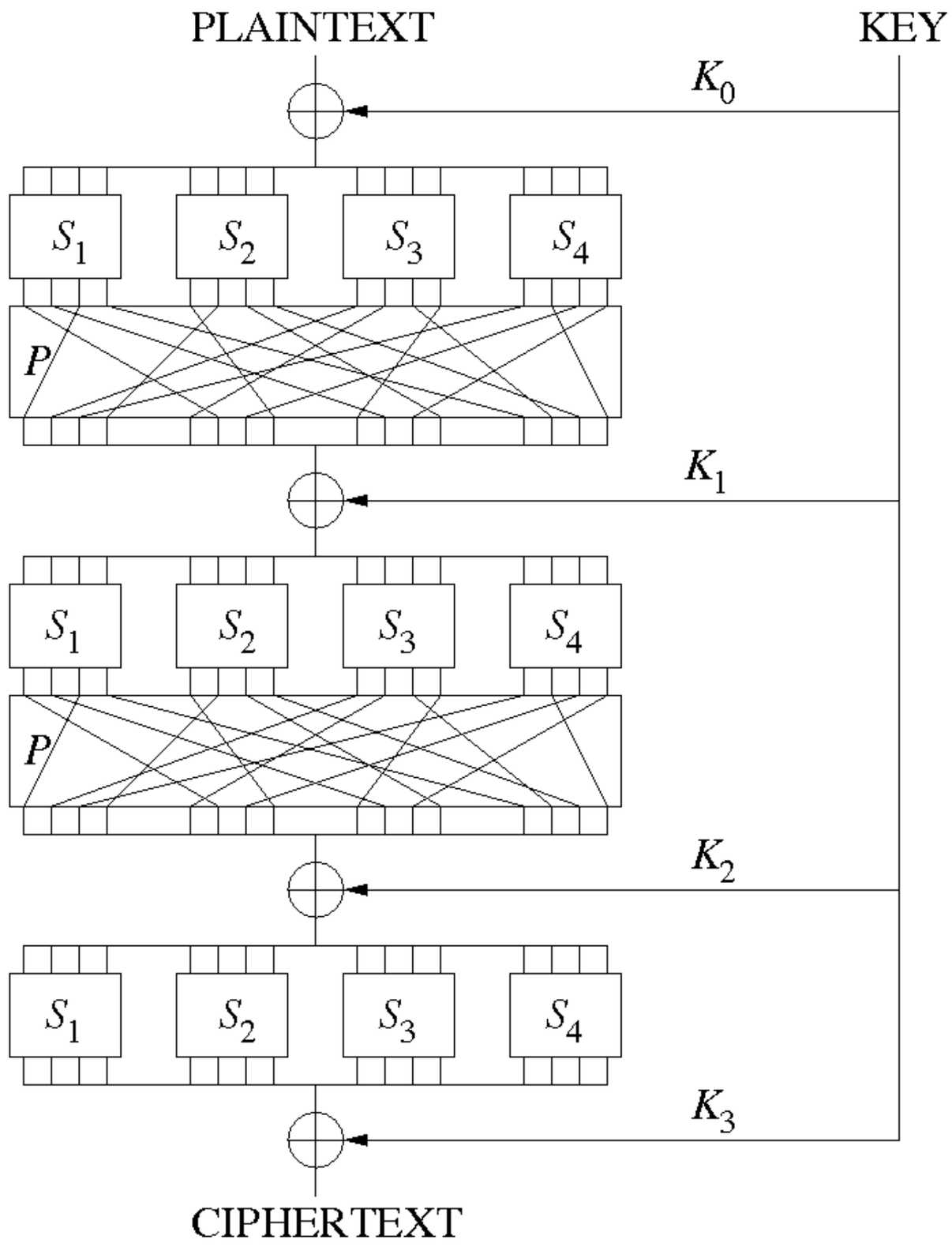
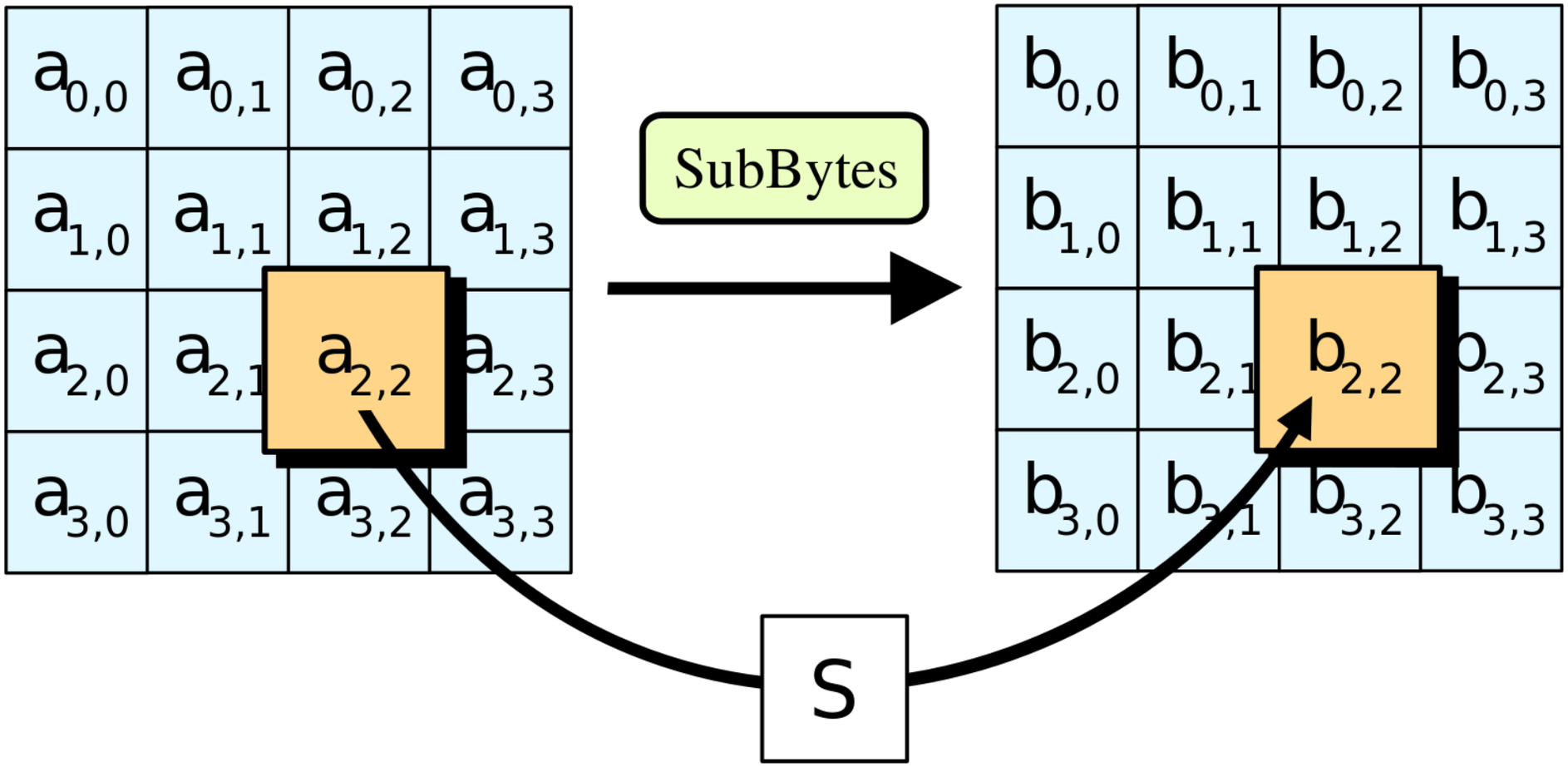
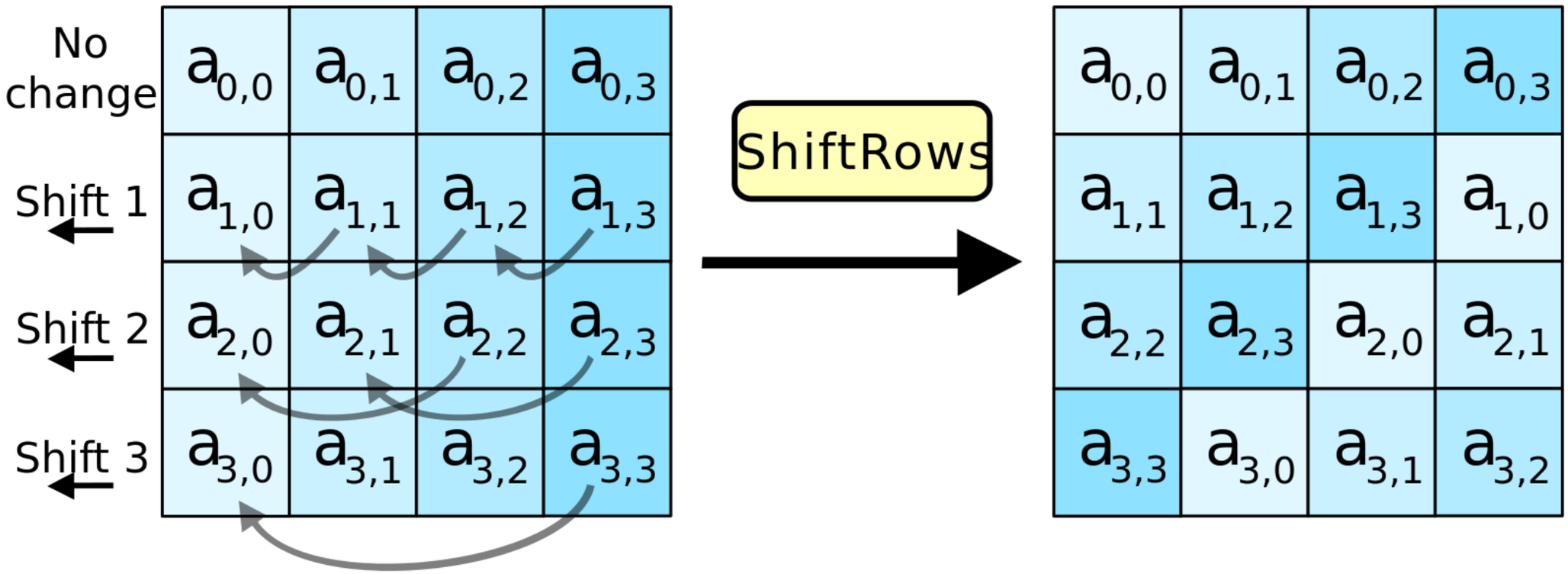
Криптография - 2

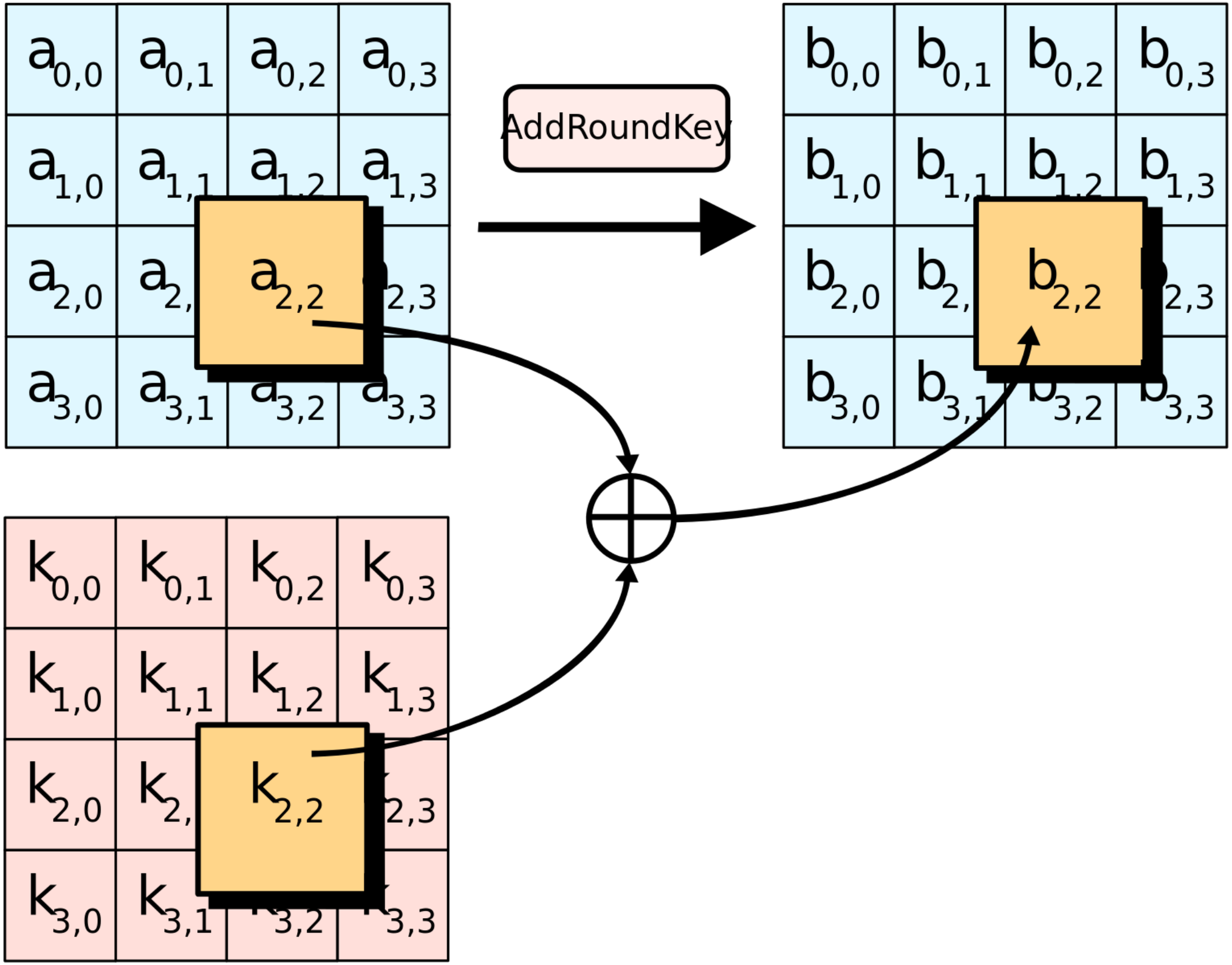
Виталий Павленко, «Интеллектуал»

Симметричное шифрование


PLAINTEXT
KEY
K_0
S_1
S_2
S_3
S_4
P
K_1
S_1
S_2
S_3
S_4
P
K_2
S_1
S_2
S_3
S_4
K_3
CIPHERTEXT

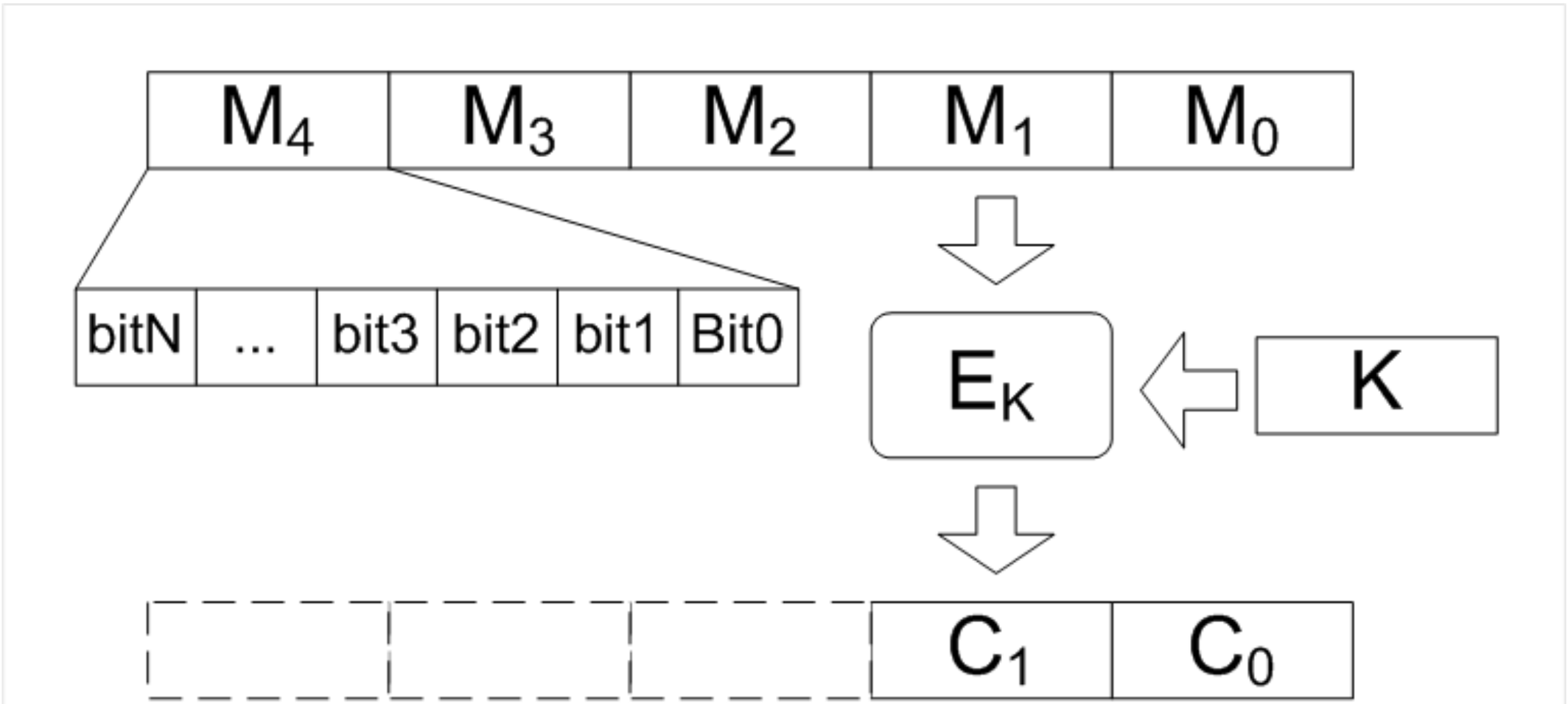

a0,0
a0,1
a0,2
a0,3
a1,0
a1,1
a1,2
a1,3
a2,0
a2,1
a2,2
a2,3
a3,0
a3,1
a3,2
a3,3
SubBytes
b0,0
b0,1
b0,2
b0,3
b1,0
b1,1
b1,2
b1,3
b2,0
b2,1
b2,2
b2,3
b3,0
b3,1
b3,2
b3,3
S

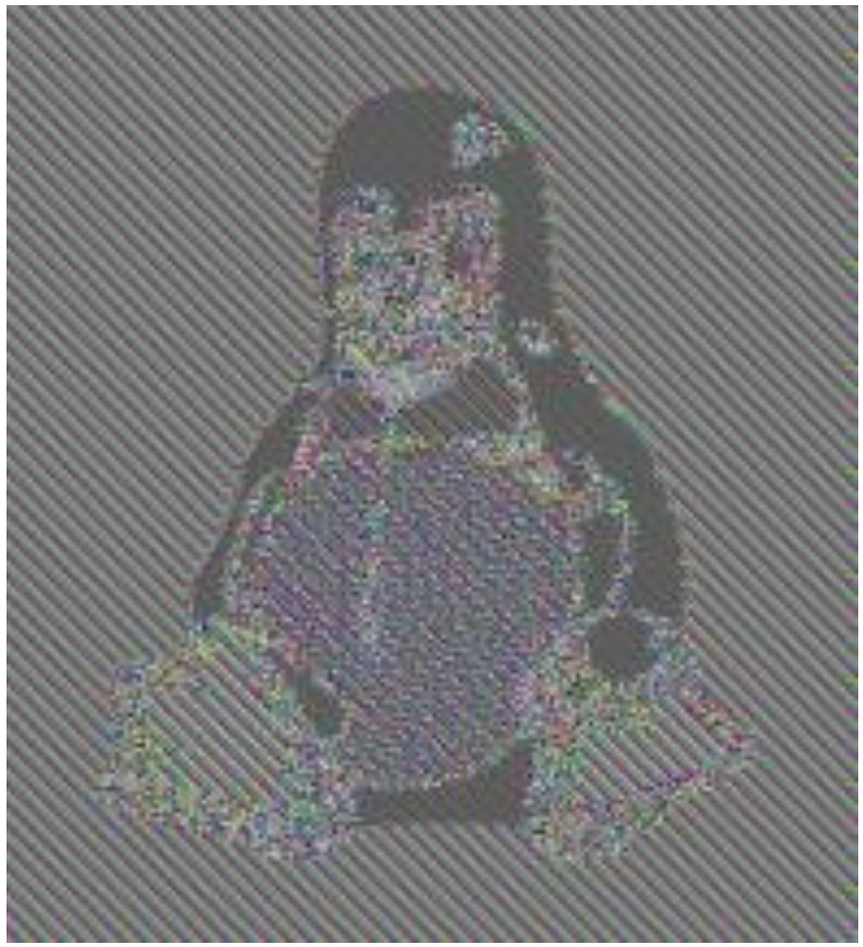

No change
Shift 1
Shift 2
Shift 3
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
ShiftRows
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,1}$ $a_{1,2}$ $a_{1,3}$ $a_{1,0}$
$a_{2,2}$ $a_{2,3}$ $a_{2,0}$ $a_{2,1}$
$a_{3,3}$ $a_{3,0}$ $a_{3,1}$ $a_{3,2}$

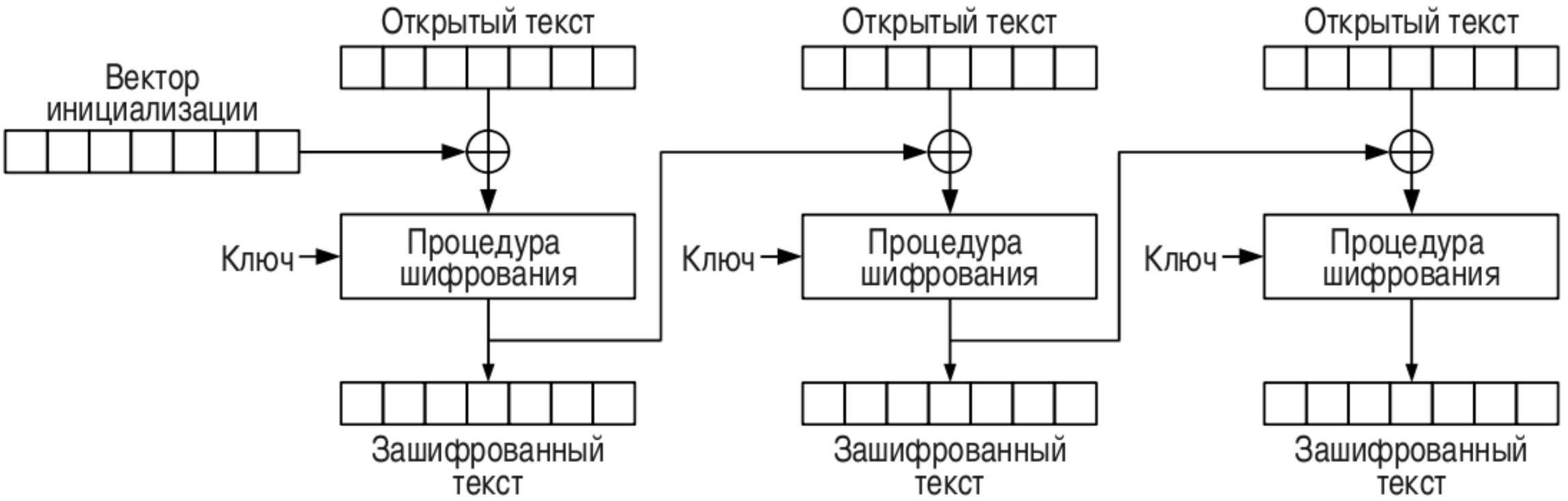

a0,0 a0,1 a0,2 a0,3
a1,0 a1,1 a1,2 a1,3
a2,0 a2,1 a2,2 a2,3
a3,0 a3,1 a3,2 a3,3
AddRoundKey
b0,0 b0,1 b0,2 b0,3
b1,0 b1,1 b1,2 b1,3
b2,0 b2,1 b2,2 b2,3
b3,0 b3,1 b3,2 b3,3
k0,0 k0,1 k0,2 k0,3
k1,0 k1,1 k1,2 k1,3
k2,0 k2,1 k2,2 k2,3
k3,0 k3,1 k3,2 k3,3

- Типичный блочный шифр: принимает 256-битный вход X, 256-битный ключ K и возвращает 256-битный выход Y

- Как использовать его для шифрования файла размером 1 Мб?


M4
M3
M2
M1
M0
bitN
...
bit3
bit2
bit1
Bit0
EK
K
C1
C0




Открытый текст
Открытый текст
Открытый текст
Вектор
инициализации
Ключ
Процедура
шифрования
Ключ
Процедура
шифрования
Ключ
Процедура
шифрования
Зашифрованный
текст
Зашифрованный
текст
Зашифрованный
текст

Хэши

Замечание: Все используемые переменные 32 бита.

Инициализация переменных:

```
h0 = 0x67452301
h1 = 0xEFCDAB89
h2 = 0x98BADCFE
h3 = 0x10325476
h4 = 0xC3D2E1F0
```

Предварительная обработка:

Присоединяем бит `'1'` к сообщению

Присоединяем `k` битов `'0'`, где `k` наименьшее число ≥ `0` такое, что длина получившегося сообщения (в битах) сравнима по модулю `512` с `448` `(length mod 512 == 448)`

Добавляем длину исходного сообщения (до предварительной обработки) как целое `64`-битное Big-endian число, в *битах*.

```
В процессе сообщение разбивается последовательно по 512 бит:
for перебираем все такие части
    разбиваем этот кусок на 16 частей, слов по 32-бита w[i], 0 <= i <= 15

    16 слов по 32-бита дополняются до 80 32-битовых слов:
    for i from 16 to 79
        w[i] = (w[i-3] xor w[i-8] xor w[i-14] xor w[i-16]) циклический сдвиг влево 1

    Инициализация хеш-значений этой части:
    a = h0
    b = h1
    c = h2
    d = h3
    e = h4
```

Основной цикл:

```
for i from 0 to 79
    if 0 ≤ i ≤ 19 then
        f = (b and c) or ((not b) and d)
        k = 0x5A827999
    else if 20 ≤ i ≤ 39
        f = b xor c xor d
        k = 0x6ED9EBA1
    else if 40 ≤ i ≤ 59
        f = (b and c) or (b and d) or (c and d)
        k = 0x8F1BBCDC
    else if 60 ≤ i ≤ 79
        f = b xor c xor d
        k = 0xCA62C1D6

    temp = (a leftrotate 5) + f + e + k + w[i]
    e = d
    d = c
    c = b leftrotate 30
    b = a
    a = temp
```

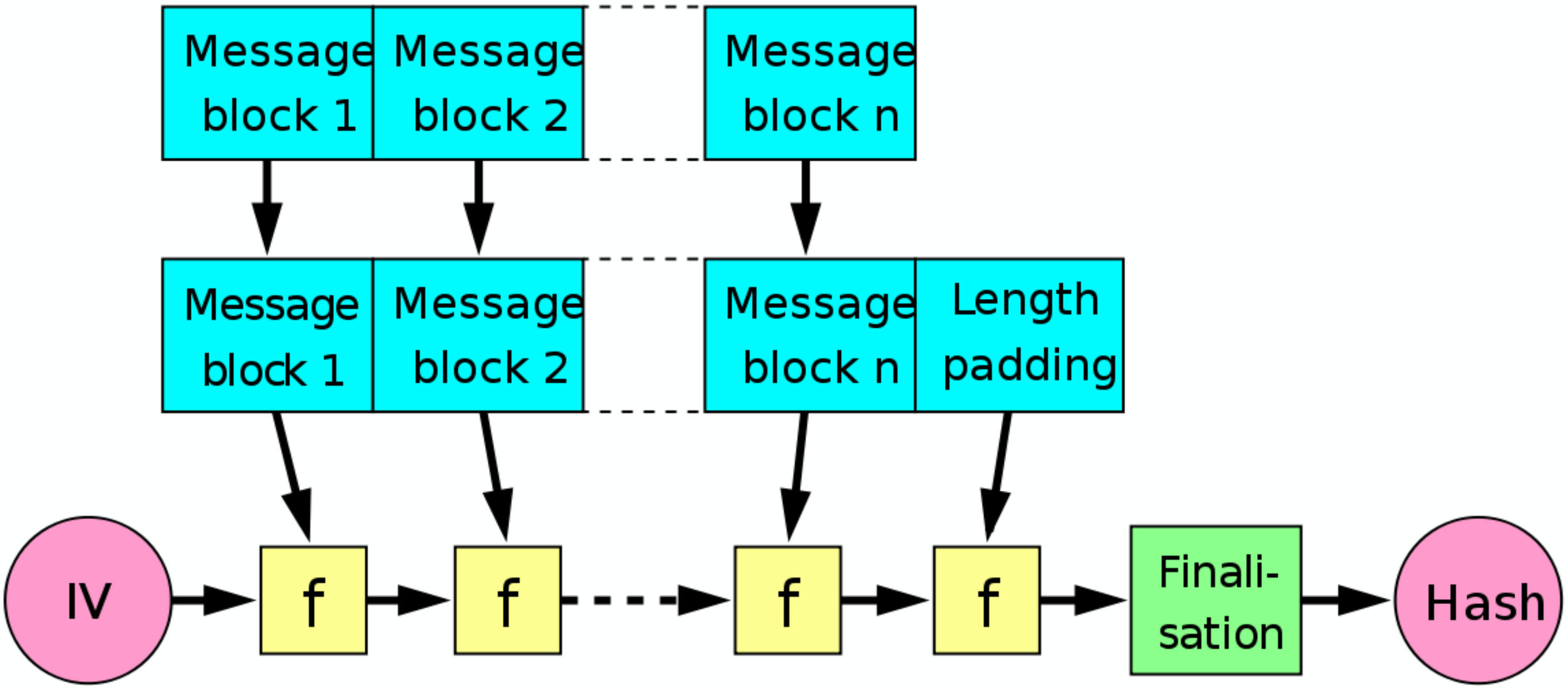
Добавляем хеш-значение этой части к результату:

```
h0 = h0 + a
h1 = h1 + b
h2 = h2 + c
h3 = h3 + d
h4 = h4 + e
```

Итоговое хеш-значение:

```
digest = hash = h0 append h1 append h2 append h3 append h4
```

- Как сделать цифровую подпись?


Message block 1
Message block 2
Message block n
Message block 1
Message block 2
Message block n
Length padding
IV
f
f
f
f
Finali-sation
Hash

Length-extension attack

```
Original Data: count=10&lat=37.351&user_id=1&long=-119.827&waffle=eggo
Original Signature: 6d5f807e23db210bc254a28be2d6759a0f5f5d99
```

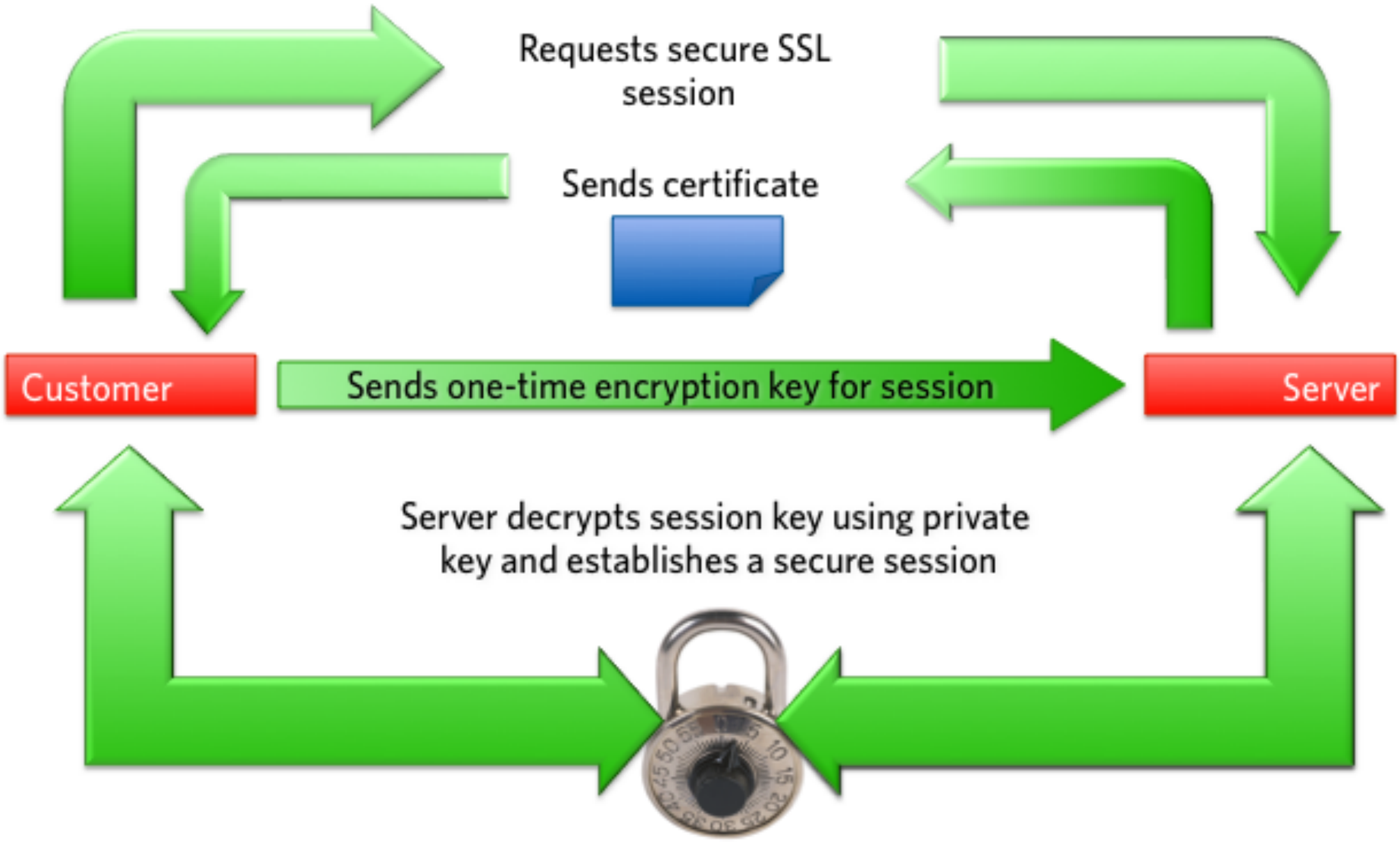
```
Desired New Data: count=10&lat=37.351&user_id=1&long=-119.827&waffle=eggo&waffle=liege
```

```
New Data: count=10&lat=37.351&user_id=1&long=-119.827&waffle=eggo\x80\x00\x00
          \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
          \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
          \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
          \x00\x00\x00\x02(&waffle=liege
```

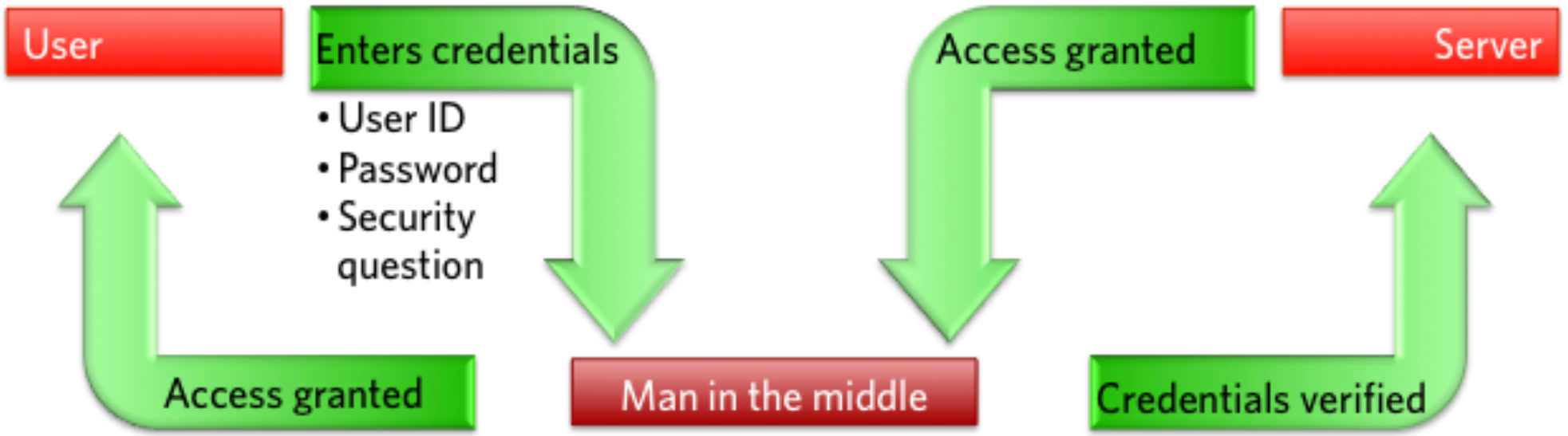
- MD5 не надёжен: люди научились находить коллизию
- fastcoll умеет генерировать два разных файла с одинаковым хэшем (содержимое вам не подвластно)
- Как сделать две таких программы: одно делает print(‘Protected’), другое print(‘Cracked’), а хэши программ одинаковые?

HTTPS

How SSL works



Man in the middle (MITM) attack



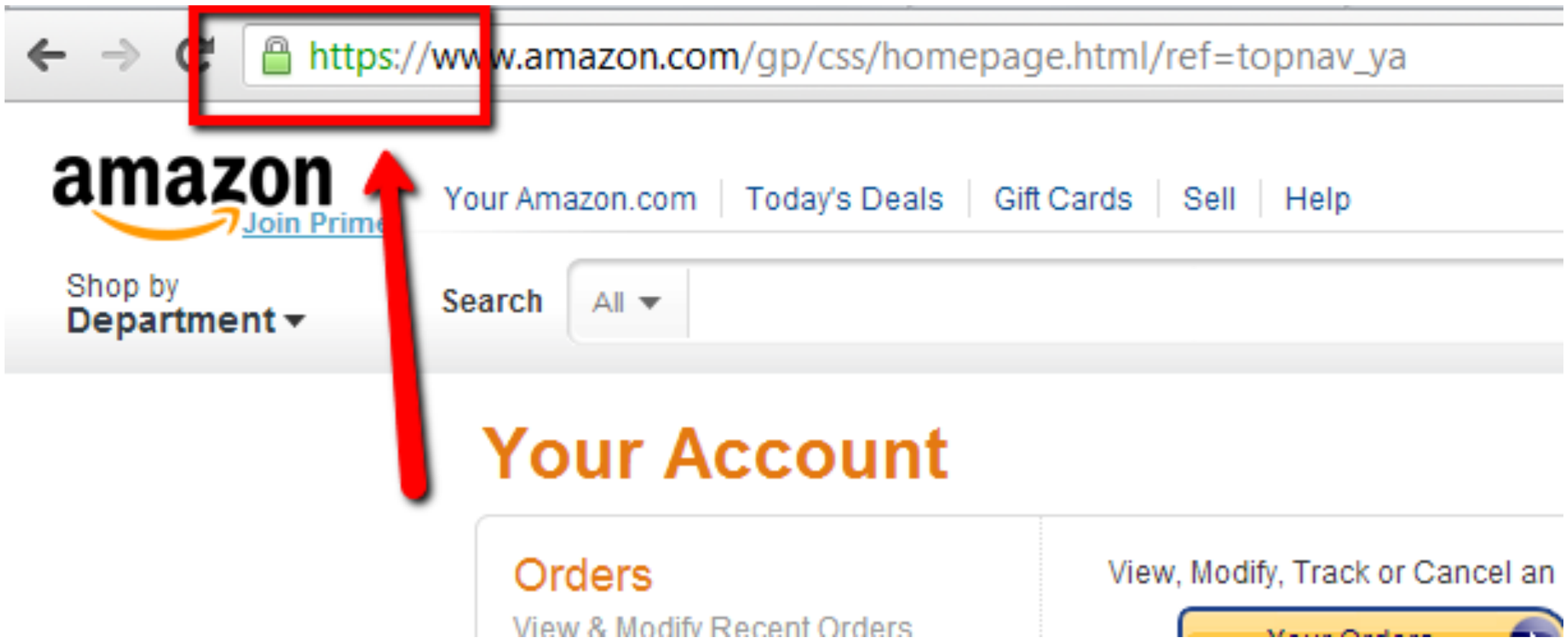
This is probably not the site you are looking for!

You attempted to reach but instead you actually reached a server identifying itself as This may be caused by a misconfiguration on the server or by something more serious. An attacker on your network could be trying to get you to visit a fake (and potentially harmful) version of

You should not proceed, **especially** if you have never seen this warning before for this site.

Proceed anyway | Back to safety

Help me understand


https://www.amazon.com/gp/css/homepage.html/ref=topnav_ya
amazon
Join Prime
Your Amazon.com
Today's Deals
Gift Cards
Sell
Help
Shop by
Department
Search
All
Your Account
Orders
View & Modify Recent Orders
View, Modify, Track or Cancel an

Certificate

General | Details | Certification Path

Certificate Information

This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer
- Proves your identity to a remote computer

* Refer to the certification authority's statement for details.

Issued to: www.amazon.com

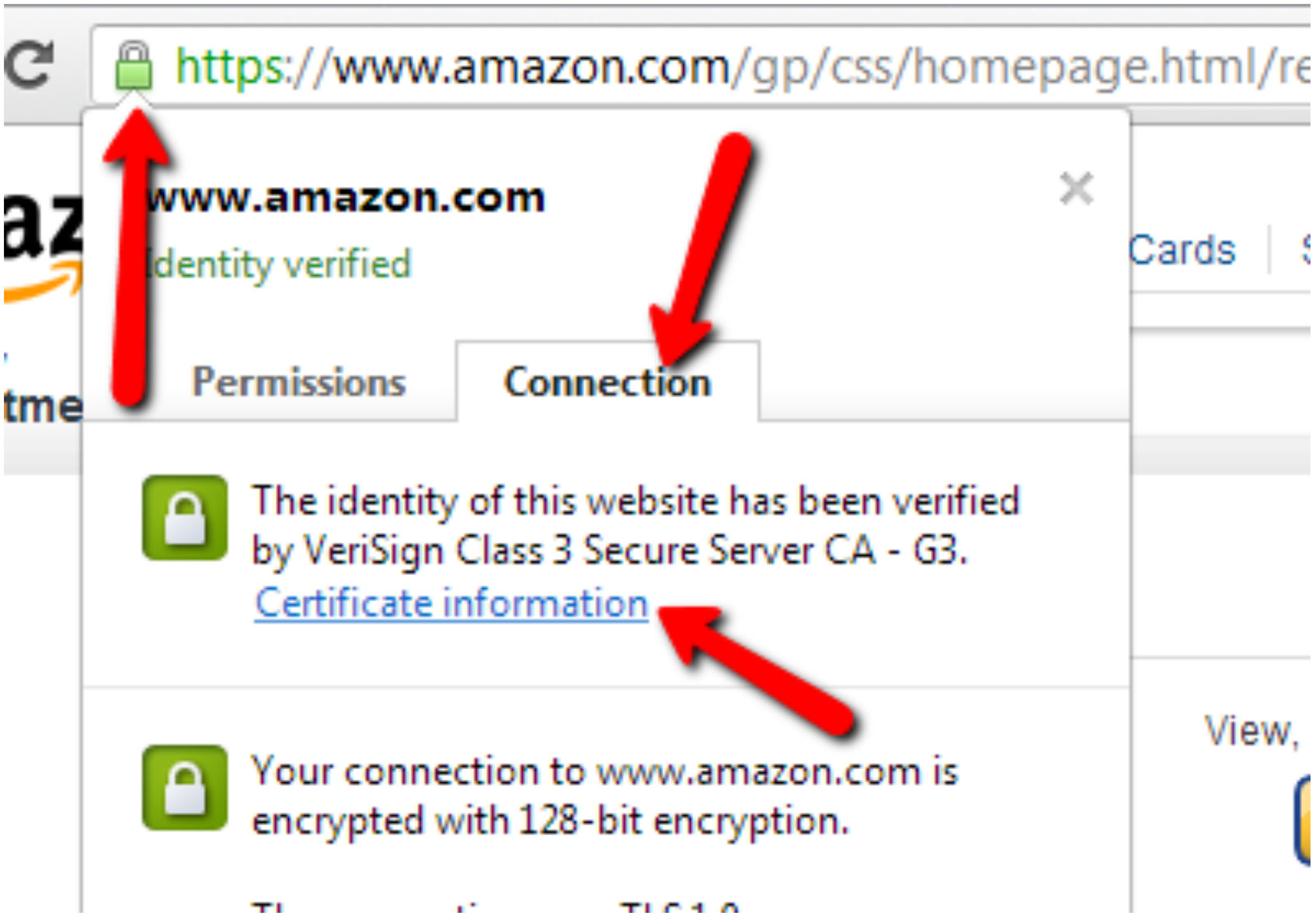
Issued by: VeriSign Class 3 Secure Server CA - G3

Valid from 16/ 05/ 2013 **to** 18/ 05/ 2014

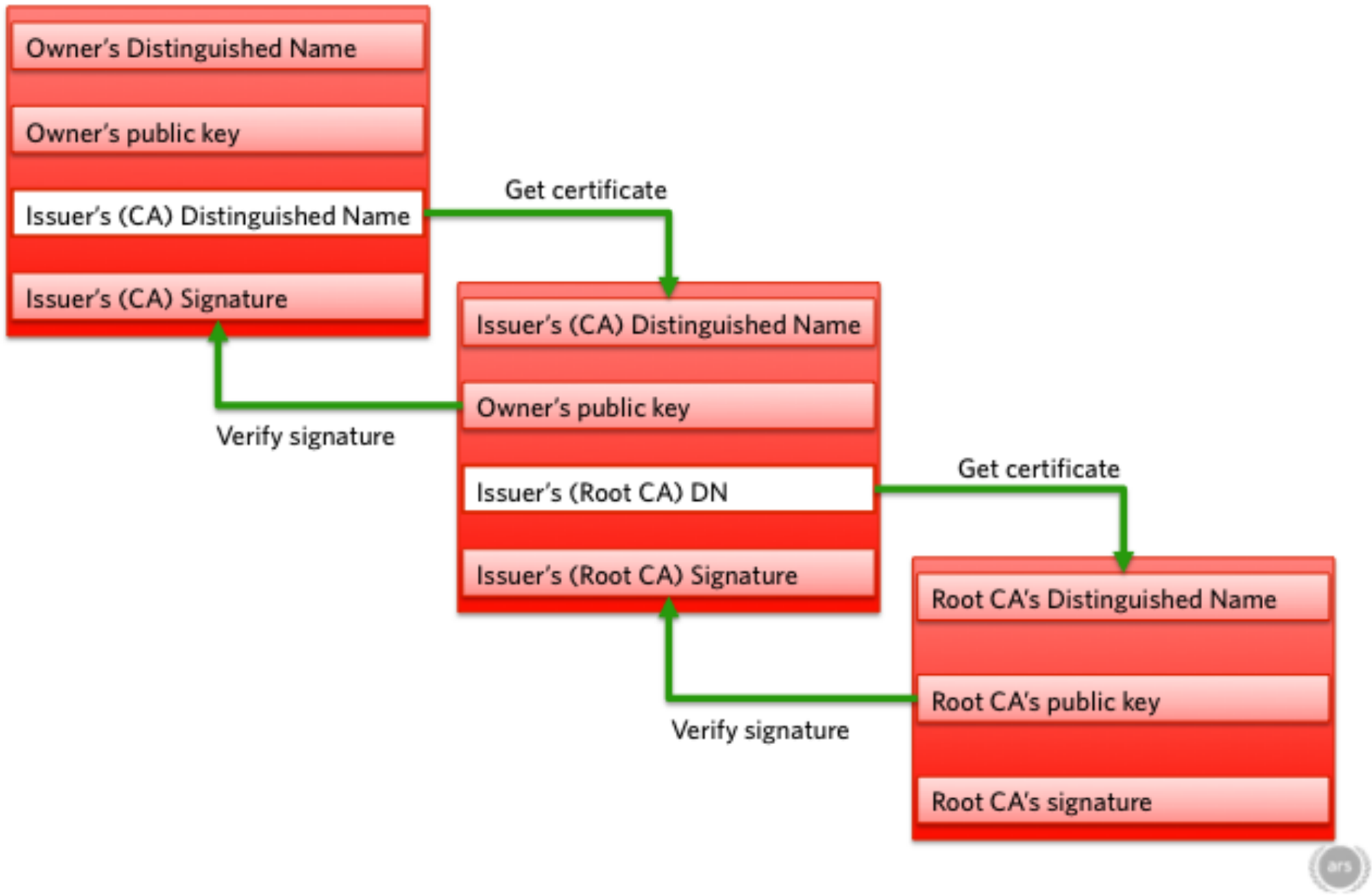
Issuer Statement

Learn more about certificates

OK


https://www.amazon.com/gp/css/homepage.html/re
www.amazon.com
Identity verified
Permissions
Connection
Cards
The identity of this website has been verified by VeriSign Class 3 Secure Server CA - G3.
Certificate information
Your connection to www.amazon.com is encrypted with 128-bit encryption.
View,

Chain of trust



Анекдоты

Из книги Д. Склярова «Искусство защиты и взлома информации»

Взлом 128-битового шифрования в Netscape

Компания Netscape разработала протокол SSL и реализовала его в своем браузере. Данные, передаваемые посредством SSL, зашифровывались алгоритмом RC4 со 128-битовым ключом. 17 сентября 1995 года Йен Голдберг (Ian Goldberg) сообщил о том, что ему в сотрудничестве с Дэвидом Вагнером (David Wagner) удалось обнаружить уязвимость в процедуре выбора 128-битового ключа для алгоритма RC4. Недостаток процедуры заключался в том, что начальное состояние генератора псевдослучайных чисел основывалось на трех значениях: идентификаторе процесса, генерирующего ключ, идентификаторе его родительского процесса и текущем времени. Учитывая то, что значительную часть информации о номерах процессов и времени можно было предугадать, пространство возможных ключей сократилось с 2^{128} до 2^{20}, и на поиск ключа шифрования уходило всего 25 секунд.

Письмо "Вашингтонского снайпера"

В октябре 2002 года в Вашингтоне и окрестностях действовал снайпер, убивавший людей без видимой причины. 24 октября был произведен арест двух подозреваемых, и убийства прекратились.

26 октября 2002 года газета "The Washington Post" выложила на своем сайте отсканированную копию письма "Вашингтонского снайпера", представленную в формате PDF. В письме, среди прочего, описывалась процедура, при помощи которой снайпер собирался получить 10 миллионов долларов. Правительство должно было перечислить деньги на счет платиновой карты Visa.

Письмо содержало все необходимые атрибуты счета и имя женщины, у которой была украдена карта. Также в нем фигурировало 3 телефонных номера. "The Washington Post" отретушировала PDF-документ, скрыв персональную информацию от публичного распространения. В результате, вместо некоторых фрагментов текста появились черные прямоугольники.

Однако метод уничтожения информации был выбран неверно. При выводе на печать все выглядело именно так, как и было задумано: прямоугольники на месте текста. Но даже при отображении на экран в программе Acrobat Reader (особенно на медленных компьютерах) можно было заметить, что сначала появляется текст, а затем поверх него прорисовываются черные заплатки. То есть персональная информация оказалась просто прикрыта, но не уничтожена.

Действительно, прямоугольники, перекрывающие изображение, легко удалить при помощи инструмента TouchUp Object Tool, входящего в коммерческую программу Adobe Acrobat. И полный текст письма "Вашингтонского снайпера" чуть ли не в тот же день можно было найти в Интернете.

- Ломается ли хитрый шифр, если неизвестен ключ?