Эссе по курсу "Защита информации", кафедра радиотехники, Московский физико-технический институт (ГУ МФТИ), http://www.re.mipt.ru/infsec

Operation modes (ECB, CBC, OFB, CFB) initialization vector IV

Хроменков Александр Васильевич

22 апреля 2007 г.

Режимы работы блочного шифра

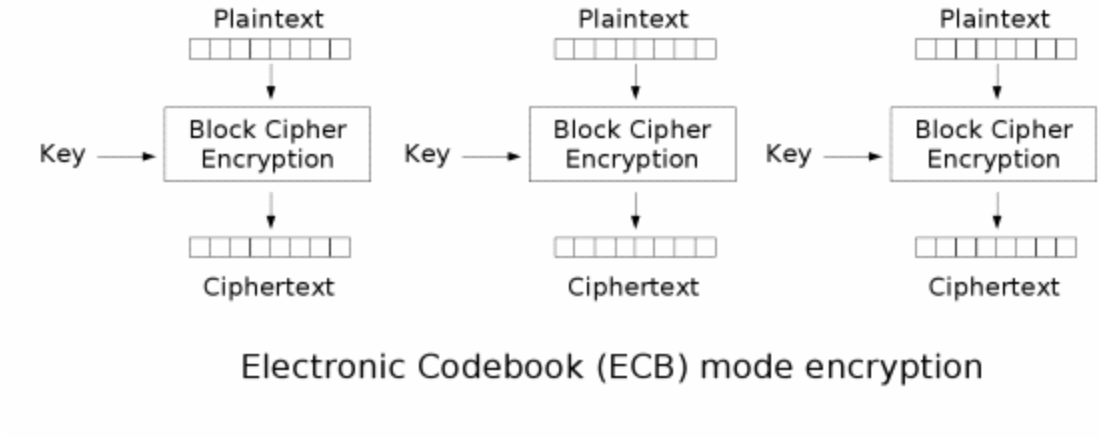
В шифровании, блочный шифр работает на блоках определенной длины, часто 64 или 128 битов. Поскольку сообщения могут иметь любую длину, и потому что шифровка того же самого открытого текста под тем же самым ключом всегда производит тот же самый вывод (как описано в разделе блока управления событием ниже), несколько **режимов работы** были изобретены, которые позволяют блочным шифрам обеспечивать конфиденциальность для сообщений произвольной длины. Самые ранние режимы, описанные в литературе (т.е блоке управления событием, CBC, OFB и CFB) обеспечивают только конфиденциальность, и не гарантируют целостность сообщения. Другие режимы были с тех пор проектированы, которые гарантируют и конфиденциальность, и целостность сообщения, типа режима CCM, режима электронного коммутатора, режима GCM, и режима OCB. Режим (LRW) узкого блочного шифрования Tweakable, и широкое блочное шифрование (канал связи и электромагнитная обстановка) режимы, проектированные, чтобы надежно зашифровать сектора диска, описаны в статье, посвященной дисковой теории кодирования.

Вектор инициализации (IV)

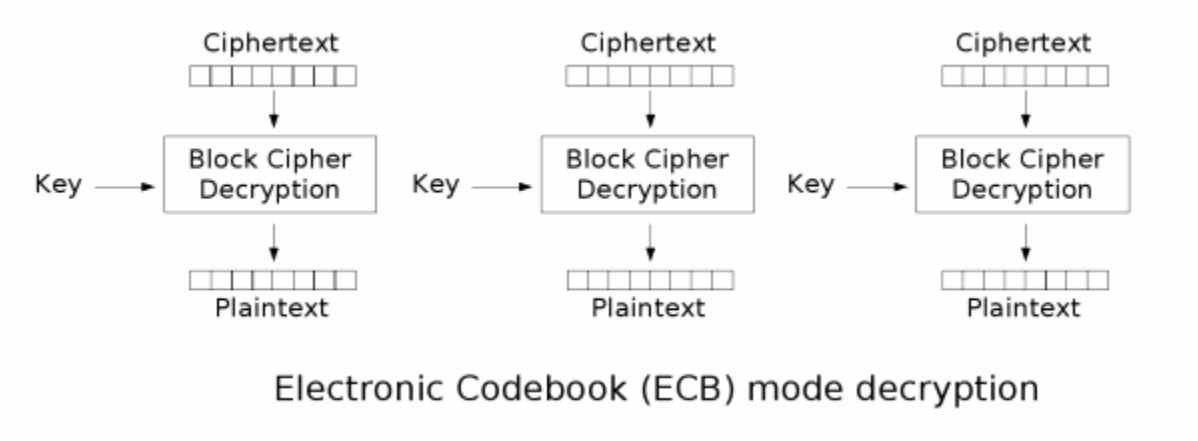
Все эти режимы (кроме блока управления событием) требуют, чтобы *вектор инициализации*, или *IV* - своего рода 'фиктивный блок' начал процесс для первого реального блока, и также обеспечил некоторую рандомизацию для процесса. Нет никакой необходимости в IV, чтобы он был секретным, в большинстве случаев, но важно, что вектор никогда не многократно использовался с тем же самым ключом. Для OFB и CTR, многократно используя IV, полностью разрушается безопасность. В режиме CBC, эти IV должны, кроме того, быть беспорядочно сгенерированы во время кодирования.

Электронная книга шифров (ECB)

Самые простые из режимов кодирования - **электронная книга шифров** (ECB). Сообщение разделено на блоки, и каждый блок зашифрован отдельно. Недостаток этого метода в том, что идентичные блоки открытого текста зашифрованы в идентичные блоки зашифрованного текста; таким образом, это не скрывает образцы данных достаточно хорошо. В некоторых смыслах, это не обеспечивает серьезную конфиденциальность сообщения, и это не рекомендуется для использования в криптографических протоколах.



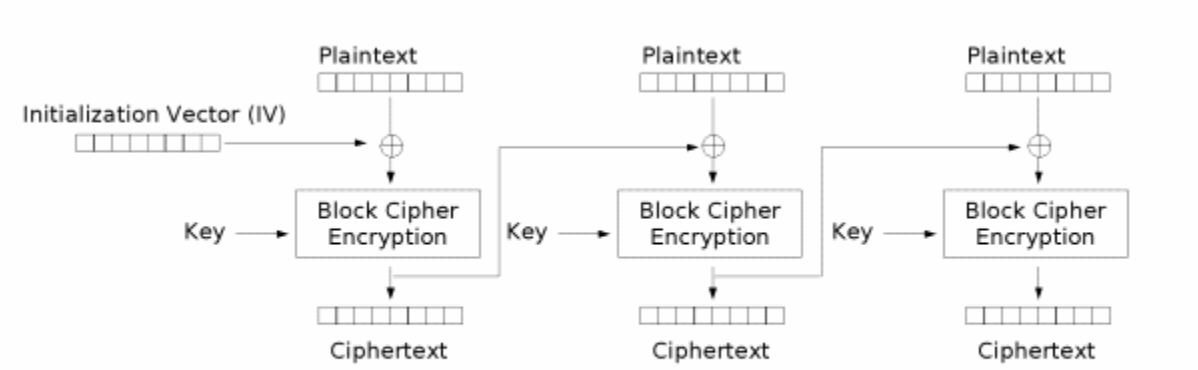
Electronic Codebook (ECB) mode encryption



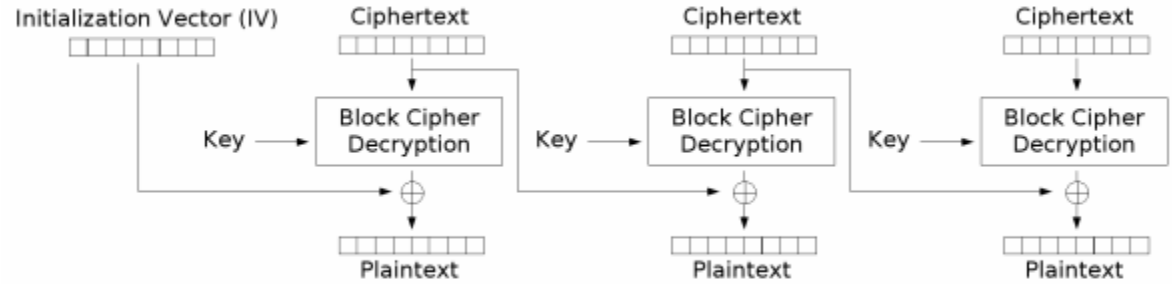
Electronic Codebook (ECB) mode decryption

Сцепление блоков шифра (CBC)

В **сцеплении блоков шифра** (CBC), каждый блок открытого текста - складывается с предыдущим блоком зашифрованного текста прежде, чем быть зашифрованным. Таким образом, каждый блок зашифрованного текста зависит от всех блоков открытого текста, обработанных до этого пункта. Кроме того, чтобы сделать каждое сообщение уникальным, вектор инициализации должен использоваться в первом блоке.



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

Если первый блок имеет индекс 1, математическая формула для кодирования CBC

$$C_i = E_K(P_i \oplus C_{i-1}), C_0 = IV$$

в то время как математическая формула для расшифровки CBC

$$P_i = D_K(C_i) \oplus C_{i-1}, C_0 = IV$$

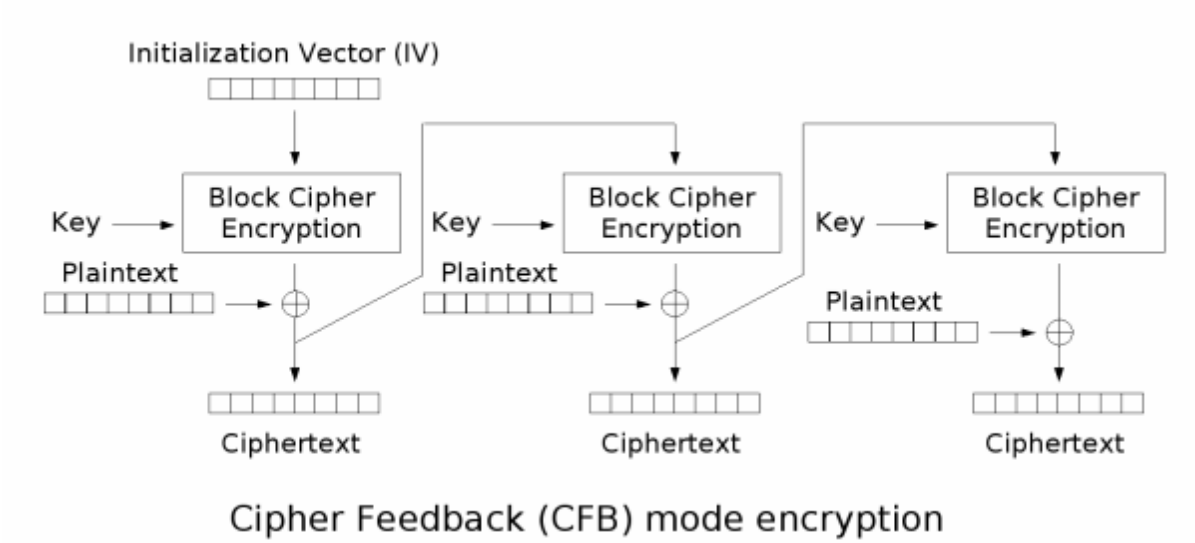
Его основные недостатки состоят в том, что кодирование является последовательным, и что сообщение должно быть дополнено к кратному числу размера блока шифра. Один способ обрабатывать эту последнюю проблему - метод, известный как захват зашифрованного текста.

Отметим, что однобитовое изменение в открытом тексте затрагивает весь после блоков зашифрованного текста, и открытый текст может быть восстановлен только от двух смежных блоков зашифрованного текста. Как следствие, в расшифровке можно найти что-либо подобное, и однобитовое изменение зашифрованного текста вызовет полное искажение соответствующего блока открытого текста, и инвертирует соответствующий бит в следующем блоке открытого текста.

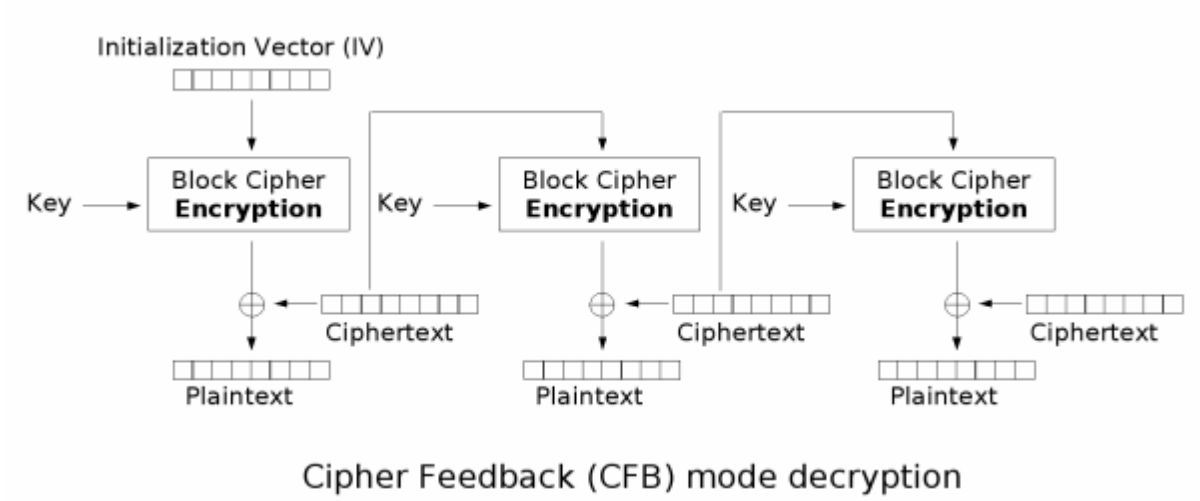
Обратная связь шифра (CFB)

Режим (CFB) **обратной связи шифра**, близкий родственник CBC, превращает блочный шифр в самосинхронизирующийся шифр потока. Операция очень похожа на предыдущую; в частности расшифровка CFB почти идентична расшифровке CBC, выполненной наоборот:

$$C_i = E_K(C_{i-1}) \oplus P_i$$
$$P_i = E_K(C_{i-1}) \oplus C_i$$
$$C_0 = \text{IV}$$



Cipher Feedback (CFB) mode encryption



Cipher Feedback (CFB) mode decryption

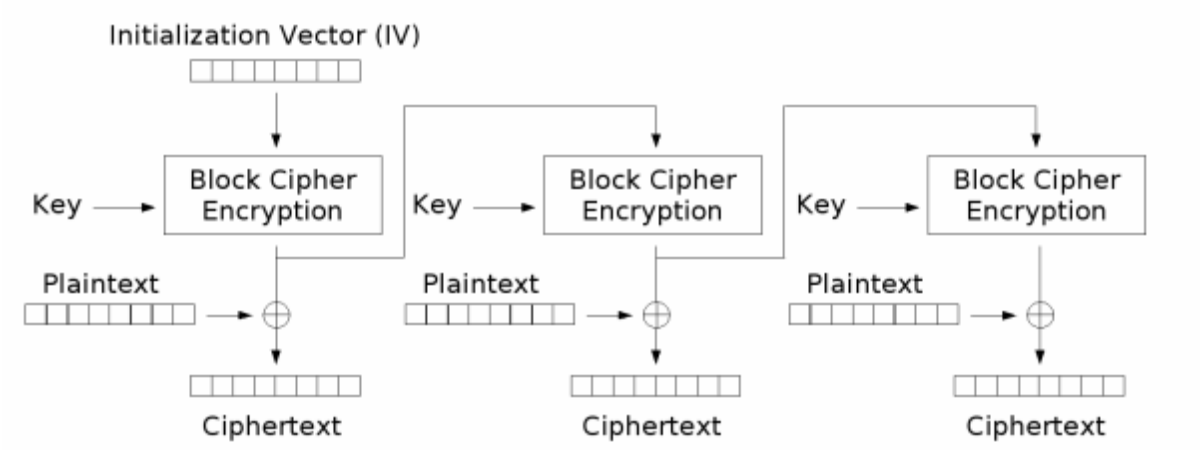
Как режим CBC, изменения в открытом тексте распространяется повсюду в зашифрованном тексте, и в кодировании нельзя найти что-либо подобное. Также как и в CBC, в расшифровке можно найти что-либо подобное. Расшифровывая, однобитовое изменение в зашифрованном тексте затрагивает два блока открытого текста: однобитовое изменение в соответствующем блоке открытого текста, и полном искажении следующего блока открытого текста. Более поздние блоки открытого текста будут расшифрованы как обычно.
Поскольку каждая стадия режима CFB зависит от зашифрованного значения предыдущего зашифрованного текста операцией сложения с текущим значением открытого текста, форма конвейерной обработки возможна, начиная с единственного шага кодирования, который требует, чтобы открытый текст был конечный операцией XOR. Это полезно для приложений, которые требуют малого времени ожидания между прибытием открытого текста и вывода соответствующего зашифрованного текста, типа определенных приложений потоковых мультимедиа.
CFB совместно использует два преимущества перед режимом CBC с режимами OFB шифра потока и CTR: блочный шифр только когда-либо используется в направлении шифровки, и сообщение не должно быть дополнено к кратному числу размера блока шифра.

Обратная связь вывода (OFB)

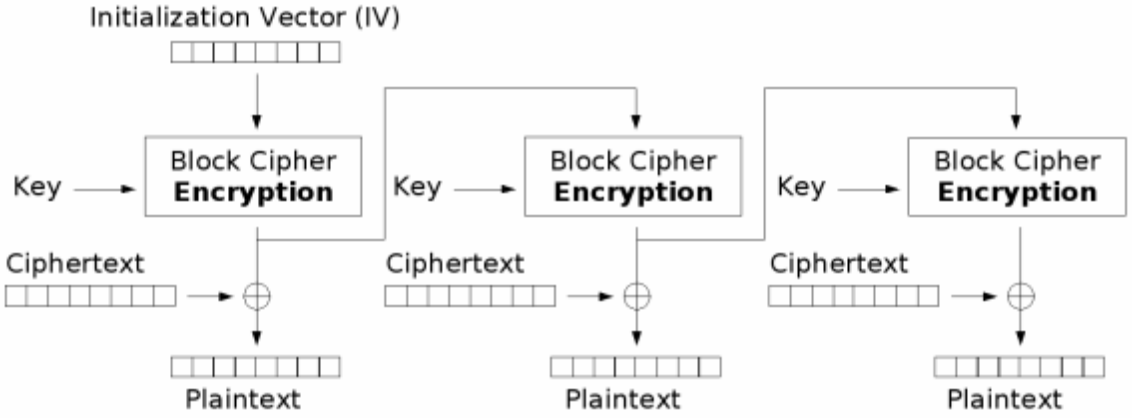
Режим (OFB) **обратной связи вывода** превращает блочный шифр в синхронный шифр потока: это генерирует ключевые блоки, которые являются результатом сложения с блоками открытого текста, чтобы получить зашифрованный текст. Так же, как с другими шифрами потока, зеркальное отражение в зашифрованном тексте производит зеркально отраженный бит в открытом тексте в том же самом местоположении. Это свойство позволяет многим кодам с исправлением ошибок функционировать как обычно, даже когда исправление ошибок применено перед кодированием.

Из-за симметрии операции сложения, кодирование и расшифровка похожи:

$$
\begin{aligned}
C_i &= P_i \oplus O_i \\
P_i &= C_i \oplus O_i \\
O_i &= E_K(O_{i-1}) \\
O_0 &= \mathrm{IV}
\end{aligned}
$$



Output Feedback (OFB) mode encryption



Output Feedback (OFB) mode decryption

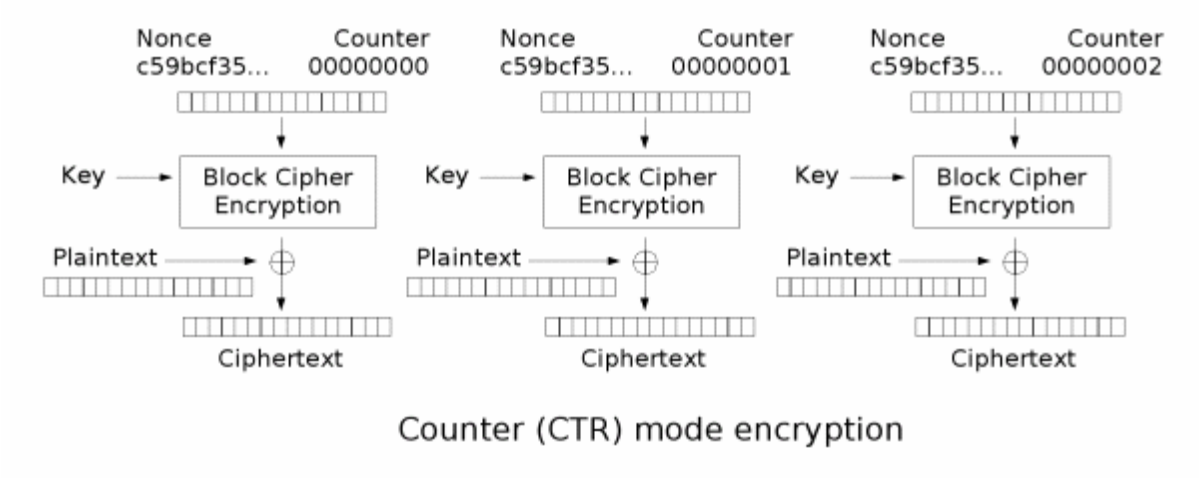
Каждая операция блочного шифра обратной связи вывода зависит от всех предыдущих, и так не может быть выполнена параллельно. Однако, потому что открытый текст или зашифрованный текст используются только для конечного сложения, операции блочного шифра могут быть выполнены заранее, позволяя выполнить заключительное шифрование параллельно с открытым текстом.

Счетчик (CTR)

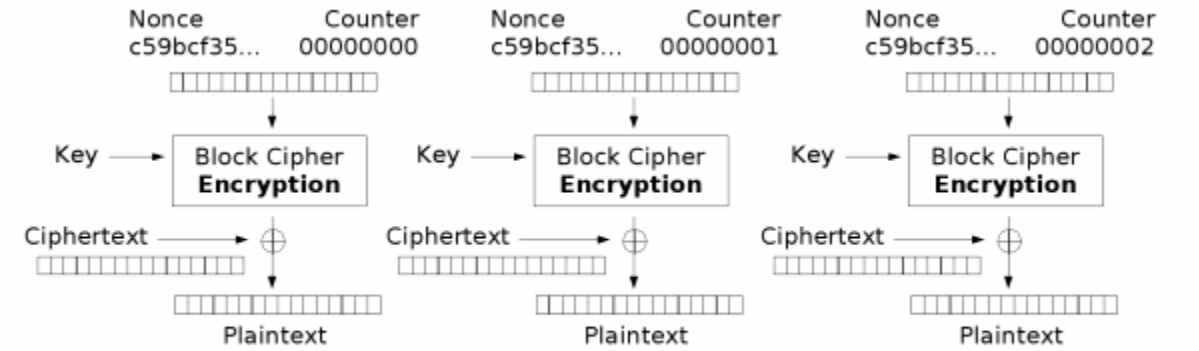
> *Примечание: Режим CTR также известен как Сегментированный Целочисленный Встречный режим (SIC)*

Как OFB, встречный режим превращает блочный шифр в шифр потока. Это генерирует следующий блок keystream, зашифровав последовательные значения "счетчика". Счетчик может быть любой простой функцией, которая производит последовательность, которая, как гарантируют, не повторится в течение долгого времени, хотя фактический счетчик является самым простым и самым популярным. Режим CTR имеет подобные характеристики к OFB, но также и позволяет использовать свойство произвольного доступа в течение расшифровки, и как полагают, является столь же безопасным как используемый блочный шифр. Отметьте, что данный случай в этом графе - тот же самый, как вектор инициализации (IV) в других

графах.



Counter (CTR) mode encryption



Counter (CTR) mode decryption

Однако когда используется надлежащая защита целостности, такая ошибка закончится (с высокой вероятностью) во всем изменённом сообщении. Если сопротивление случайной ошибке желательно, коды с исправлением ошибок должны быть применены к зашифрованному тексту перед передачей.
Немного режимов работы были проектированы, чтобы объединить безопасность и идентификацию. Примеры таких режимов: IACBC, IAPM, режим OCB, электронный коммутатор, CWC, CCM, и GCM. Эти заверенные режимы кодирования классифицированы как единственные режимы прохода или двойные режимы прохода.

Материалы, использованные в статье:

- http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation
- http://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B6%D0%B8%D0%BC_%D1%88%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F
- http://citforum.ru/security/cryptography/rejim_shifrov/
- http://www.connect.ru/article.asp?id=6545