Chosen ciphertext security importance

И. Жирков[1]

[1]Computer Science Center

22 февраля 2012 г.

Содержание

План

Definition (Математическая модель системы шифрования/дешифрования дискретных сообщений)

Пара функций:

$$\begin{cases} E = f(M, e) \\ D = g(E, d) \end{cases}$$

которые преобразуют сообщение M в криптограмму E при помощи ключа шифрования e и, наоборот, криптограмму E в сообщение M при помощи ключа дешифрования d .

- Функции $E = f(M, e)$ и $D = g(E, d)$ при известных аргументах вычисляются сравнительно просто.
- Функция $D = g(E, ?)$ при неизвестном ключе дешифрования вычисляется достаточно сложно.

Definition (Математическая модель системы шифрования/дешифрования дискретных сообщений)

Пара функций:

$$\begin{cases} E = f(M, e) \\ D = g(E, d) \end{cases}$$

которые преобразуют сообщение M в криптограмму E при помощи ключа шифрования e и, наоборот, криптограмму E в сообщение M при помощи ключа дешифрования d .

- Функции $E = f(M, e)$ и $D = g(E, d)$ при известных аргументах вычисляются сравнительно просто.
- Функция $D = g(E, ?)$ при неизвестном ключе дешифрования вычисляется достаточно сложно.

Definition (Математическая модель системы шифрования/дешифрования дискретных сообщений)

Пара функций:

$$\begin{cases} E = f(M, e) \\ D = g(E, d) \end{cases}$$

которые преобразуют сообщение M в криптограмму E при помощи ключа шифрования e и, наоборот, криптограмму E в сообщение M при помощи ключа дешифрования d .

- Функции $E = f(M, e)$ и $D = g(E, d)$ при известных аргументах вычисляются сравнительно просто.
- Функция $D = g(E, ?)$ при неизвестном ключе дешифрования вычисляется достаточно сложно.

Минимальная рекомендованная длина ключа — 128 бит, т.е. существует 2^{128} возможных ключей.
Перебор требует огромного времени и вычислительных ресурсов.

План

Симметричные и асимметричные системы шифрования

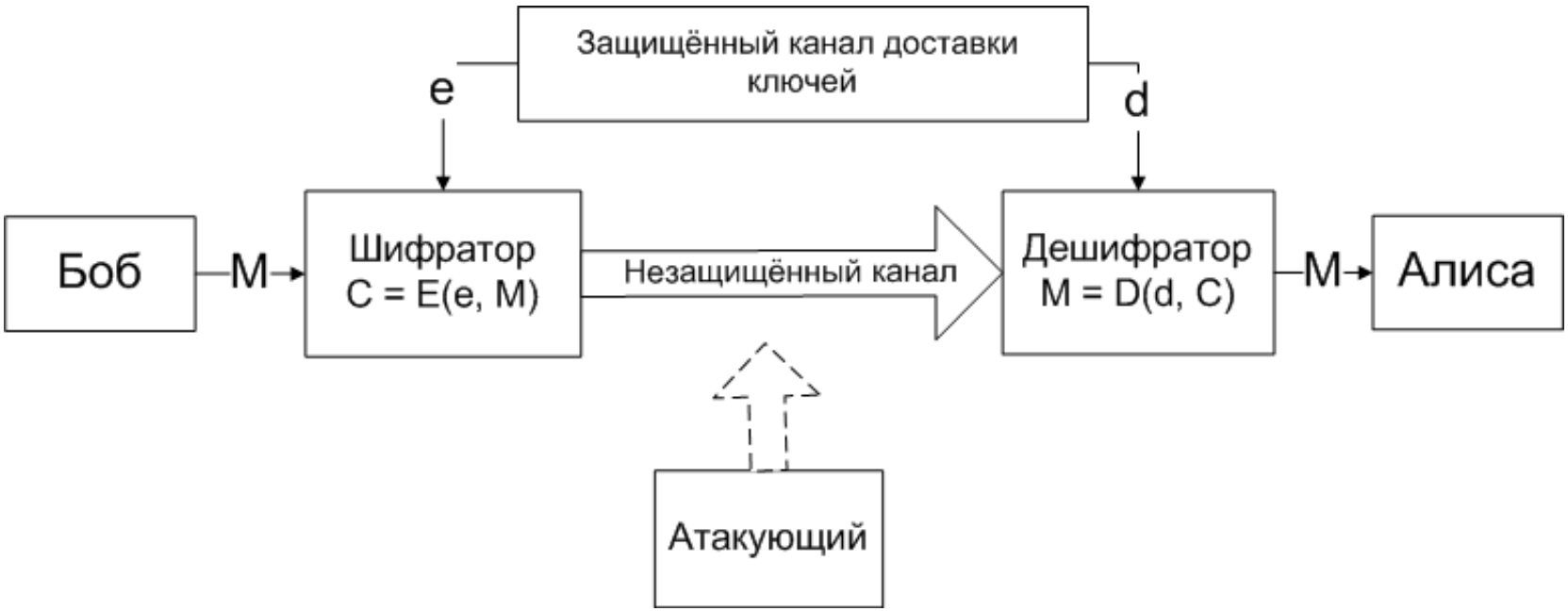
- В симметричных системах (с закрытым ключом) один из ключей (e, d) можно вычислительно просто определить по другому.
  Оба ключа должны быть секретными.
  Практически всегда $e = d$.
- В асимметричных системах (с открытым ключом) нельзя вычислительно просто получить d зная e.
  e — открытый ключ, известен даже злоумышленнику.
  Мы рассматриваем оба типа систем.

Симметричные и асимметричные системы шифрования

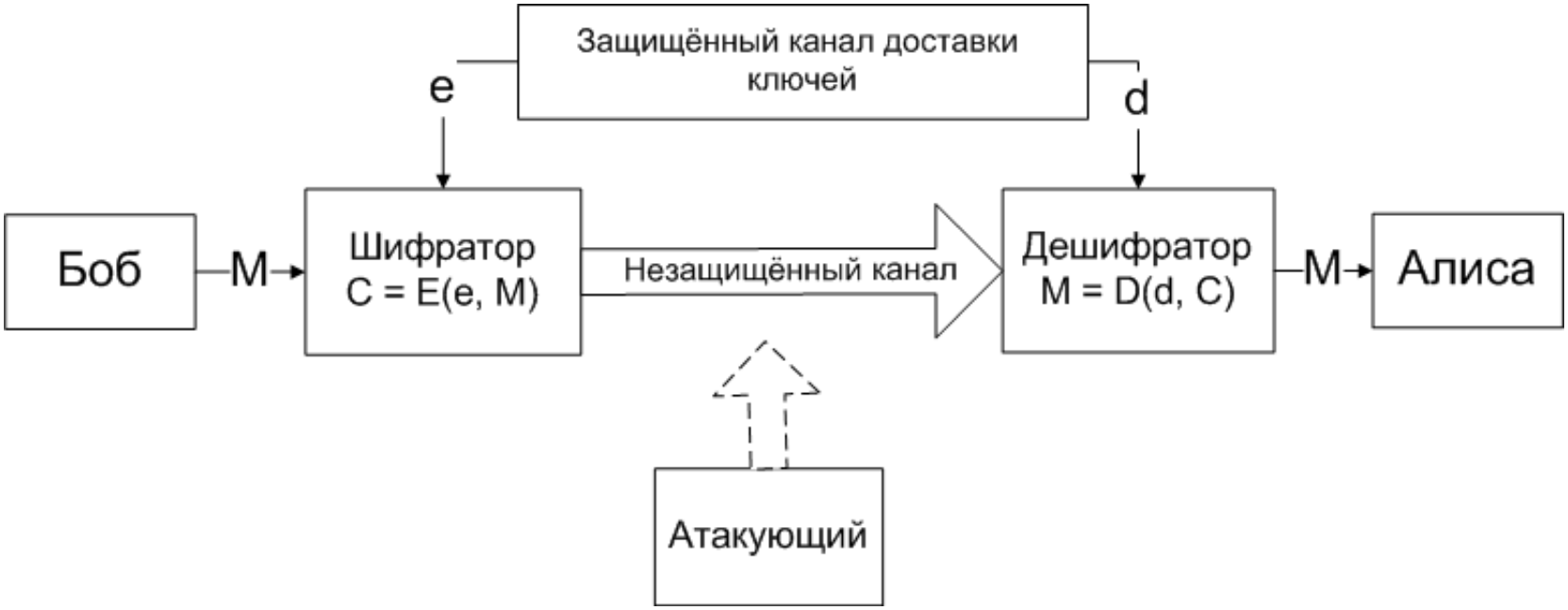
- В симметричных системах (с закрытым ключом) один из ключей (e, d) можно вычислительно просто определить по другому.
  Оба ключа должны быть секретными.
  Практически всегда $e = d$.
- В асимметричных системах (с открытым ключом) нельзя вычислительно просто получить d зная e.
  e — открытый ключ, известен даже злоумышленнику.
  Мы рассматриваем оба типа систем.

План

Модель Шеннона



Боб и Алиса — порядочные пользователи.



- Имеется источник ключей шифрования e и дешифрования d
- Шифруем сообщение с использованием e
- Передаём криптограмму по незащищённому каналу
- Получатель может восстановить сообщение с помощью дешифрующего преобразования.

План

Шеннон изучал вопросы:

- Насколько устойчива система, если криптоаналитик противника не ограничен временем и обладает всеми необходимыми средствами для анализа криптограмм?
- Имеет ли криптограмма единственное решение?
- Какой объем шифртекста необходимо перехватить криптоаналитику, чтобы решение стало единственным?

Совершенная секретность

Для любой криптограммы апостериорные вероятности равны априорным вероятностям, то есть перехват зашифрованного сообщения не дает противнику никакой информации, помогающей расшифровать его.

$$\forall C \forall M : P_C(M) = P(M)$$

В системе Шеннона противник может только перехватывать и анализировать сообщения, но не изменять их.
В качестве идеальной криптосистемы Шеннон предлагал шифрблокноты.
Не только непрактичны, но и не гарантируют целостность.

В системе Шеннона противник может только перехватывать и анализировать сообщения, но не изменять их.
В качестве идеальной криптосистемы Шеннон предлагал шифрблокноты.
Не только непрактичны, но и не гарантируют целостность.

Любое определение "безопасности" для криптосистемы состоит из двух пунктов:

- Каковы возможности злоумышленника?
- Что значит "сломать" систему?

В зависимости от возможностей злоумышленника говорят о:

- Пассивном злоумышленнике (Semantic Security)
- Активном злоумышленнике (Chosen Ciphertext Security, Non-Malleability)

Удобно определять понятия безопасности как описания антагонистических игр между "хорошим парнем" Грегом и "плохим парнем" Стивом.

План

Semantic Security

- Злоумышленник "подслушивает" (eavesdropping adversary)
- Что означает сломать Semantic Secure систему?

  1. Грег генерирует ключ.
  2. Стив даёт Грегу сообщения m_0 и m_1 одинаковой длины.
  3. Грег бросает монетку. Если выпала решка, он зашифровывает m_0 и даёт криптограмму Стиву, иначе m_1.
  4. Стив должен угадать, какое из двух сообщений зашифровал Грег.

Для идеально стойкой криптосистемы, у Стива не должно быть шанса угадать сообщение с вероятностью, большей $\frac{1}{2}$. Для вычислительно стойкой криптосистемы, — эффективного способа угадать с вероятностью существенно большей, чем $\frac{1}{2}$.

Если атакующий может только прослушивать сообщения, то Semantic Security — адекватное определение безопасности. Обычно же атакующий может играть активную роль: модифицировать сообщения или отправлять свои собственные.

Chosen Plaintext Security

- Теперь злоумышленник может попросить зашифровать произвольный текст
- 1. Грег генерирует ключ.
  2. Стив просит Грега зашифровать произвольное количество текстов и получает их криптограммы.
  3. Стив даёт Грегу сообщения m_0 и m_1 одинаковой длины.
  4. Грег бросает монетку. Если выпала решка, он шифрует m_0 и отправляет его Стиву, иначе m_1. Назовём эту криптограмму e'.
  5. Стив просит Грега зашифровать произвольное количество текстов, включая m_0 и m_1, и получает их криптограммы.
  6. Стив должен угадать, какое из двух сообщений зашифровал Грег.

Chosen Ciphertext Security

- А теперь Стив может давать криптограммы на расшифровку!
- 1. Грег генерирует ключ.
  2. Стив просит Грега расшифровать любые криптограммы и шифровать любые тексты.
  3. Стив даёт Грегу сообщения m_0 и m_1 одинаковой длины.
  4. Грег бросает монетку. Если выпала решка, он шифрует m_0 и отправляет его Стиву, иначе m_1. Назовём эту криптограмму e'.
  5. Стив может попросить Грега расшифровать любые криптограммы (кроме e') и шифровать любые тексты.
  6. Стив должен угадать, какое из двух сообщений зашифровал Грег.

Вопрос

- Как добрый Грег отреагирует на некорректные криптограммы Стива и заметит ли Стив его необычную реакцию?
- Какую информацию Стив может почерпнуть из реакции Грега, поможет ли она ему "сломать" систему?

Chosen Ciphertext Security — очень сильное определение.

- Сложно доказать, что система или протокол удовлетворяют ему.
- Многие инженеры полагают, что им не нужны такие сильные гарантии столь большими усилиями.

Chosen Ciphertext Security — очень сильное определение.

- Сложно доказать, что система или протокол удовлетворяют ему.
- Многие инженеры полагают, что им не нужны такие сильные гарантии столь большими усилиями.

План

Non-malleability

Параллельно с концепцией Chosen Ciphertext Security развивалась концепция Non-malleability, т.е. невозможность модификации криптограммы с предсказуемым результатом.

Было формально доказано, что свойство Non-Malleability эквивалентно Chosen Ciphertext Security.

Интуитивно: если активный злоумышленник в ходе атаки не получает информации, помогающей расшифровать сообщение, то он не может и изменить сообщение внутри криптограммы предсказуемым образом.

Non-malleability

Параллельно с концепцией Chosen Ciphertext Security развивалась концепция Non-malleability, т.е. невозможность модификации криптограммы с предсказуемым результатом. Было формально доказано, что свойство Non-Malleability эквивалентно Chosen Ciphertext Security.

Интуитивно: если активный злоумышленник в ходе атаки не получает информации, помогающей расшифровать сообщение, то он не может и изменить сообщение внутри криптограммы предсказуемым образом.

Non-malleability

Параллельно с концепцией Chosen Ciphertext Security развивалась концепция Non-malleability, т.е. невозможность модификации криптограммы с предсказуемым результатом. Было формально доказано, что свойство Non-Malleability эквивалентно Chosen Ciphertext Security.
Интуитивно: если активный злоумышленник в ходе атаки не получает информации, помогающей расшифровать сообщение, то он не может и изменить сообщение внутри криптограммы предсказуемым образом.

План

На самом деле достичь свойства Chosen Ciphertext Security можно просто добавив MAC в Chosen Plaintext Secure систему — для систем с закрытым ключом.

Definition (MAC)

MAC это набор алгоритмов:

- Gen — генерирует ключ на основании своего входа
- Mac генерирует тэг на основании ключа и сообщения. Бывает рандомизирован.
- Vrfy на основании ключа, сообщения и тэга заключает, корректно ли сообщение.

Интуитивно: Криптографическая контрольная сумма (как контрольная сумма, но проверить её могут только хорошие парни), или симметричный аналог цифровой подписи.

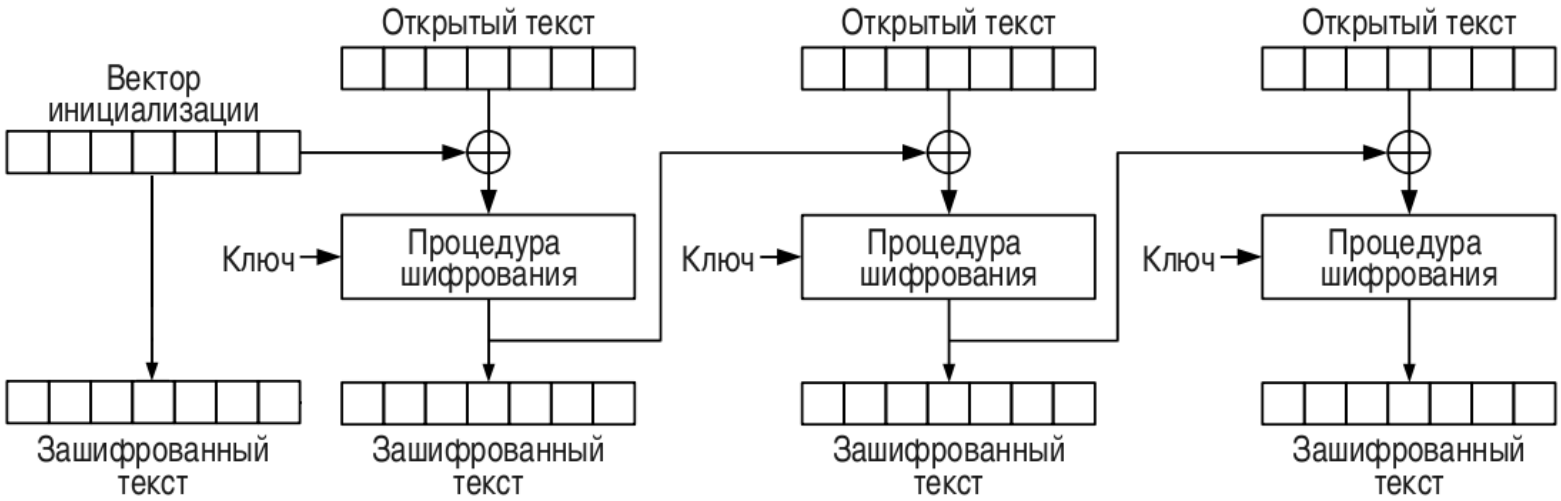
План

- В 1998 Блайхенбахер[2] выступил с сообщением об уязвимости протоколов, основанных на RSA, являющейся следствием незащищённости RSA от активных атак.
- В 2002 году Vaudenay [3] теоретически обосновал несоответствие SSL, IPSEC и некоторых других стандартов определению Chosen Ciphertext Security и предсказал возможность атак на них.
- В 2010 году Риззо и Дуонг[4] представили уязвимость шифрования в CBC режиме, которое в большинстве применений не удовлетворяет требованию Chosen Ciphertext Security, и показали практические примеры атак: взлом CAPTCHA, атаки на Apache, Ruby on Rails.

План

Cipher Block Chaining

Режим сцепления блоков шифротекста



Выравнивание

	BLOCK #1								BLOCK #2							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Ex 1	F	I	G													
Ex 1 (Padded)	F	I	G	0x05	0x05	0x05	0x05	0x05								
Ex 2	B	A	N	A	N	A										
Ex 2 (Padded)	B	A	N	A	N	A	0x02	0x02								
Ex 3	A	V	O	C	A	D	O									
Ex 3 (Padded)	A	V	O	C	A	D	O	0x01								
Ex 4	P	L	A	N	T	A	I	N								
Ex 4 (Padded)	P	L	A	N	T	A	I	N	0x08	0x08	0x08	0x08	0x08	0x08	0x08	0x08
Ex 5	P	A	S	S	I	O	N	F	R	U	I	T				
Ex 5 (Padded)	P	A	S	S	I	O	N	F	R	U	I	T	0x04	0x04	0x04	0x04

PCKS#5 — распространённый стандарт выравнивания.

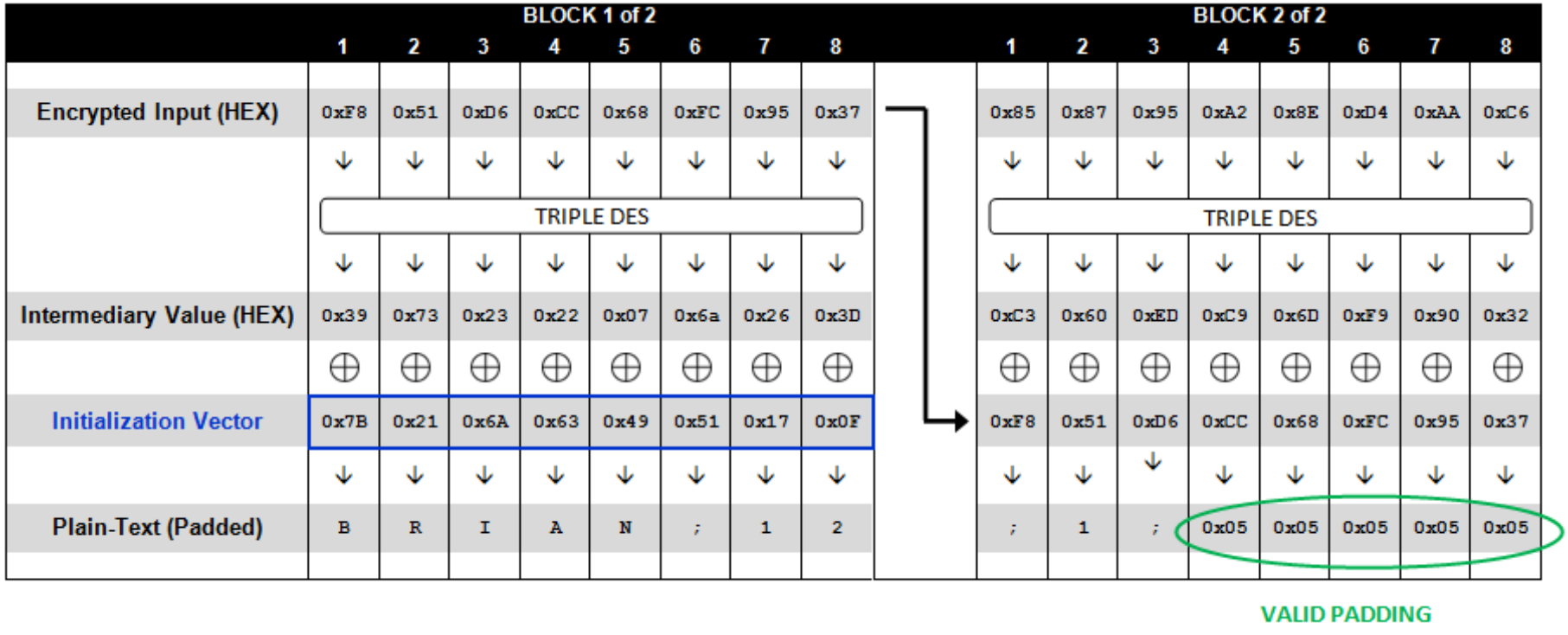
http://sampleapp/home.jsp?UID=
7B216A634951170FF851D6CC68FC9537858

	INITIALIZATION VECTOR								BLOCK 1 of 2								BLOCK 2 of 2							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Plain-Text	-	-	-	-	-	-	-	-	B	R	I	A	N	;	1	2	;	1	;					
Plain-Text (Padded)	-	-	-	-	-	-	-	-	B	R	I	A	N	;	1	2	;	1	;	0x05	0x05	0x05	0x05	0x05
Encrypted Value (HEX)	0x7B	0x21	0x6A	0x63	0x49	0x51	0x17	0x0F	0xF8	0x51	0xD6	0xCC	0x68	0xFC	0x95	0x37	0x85	0x87	0x95	0xA2	0x8E	0xD4	0xAA	0xC6

Шифрование

	BLOCK 1 of 2								BLOCK 2 of 2							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Initialization Vector	0x7B	0x21	0x6A	0x63	0x49	0x51	0x17	0x0F	0xF8	0x51	0xD6	0xCC	0x68	0xFC	0x95	0x37
	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
Plain-Text (Padded)	B	R	I	A	N	;	1	2	;	1	;	0x05	0x05	0x05	0x05	0x05
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Intermediary Value (HEX)	0x39	0x73	0x23	0x22	0x07	0x6A	0x26	0x3D	0xC3	0x60	0xED	0xC9	0x6D	0xF9	0x90	0x32
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
	TRIPLE DES								TRIPLE DES							
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Encrypted Output (HEX)	0xF8	0x51	0xD6	0xCC	0x68	0xFC	0x95	0x37	0x85	0x87	0x95	0xA2	0x8E	0xD4	0xAA	0xC6

Расшифровка



План

- Берём первый блок криптограммы (не IV);
- Берем IV состоящий из нулей;
- Посылаем приложению запрос в виде конкатенации нулевого IV и первого блока криптограммы

UID=0000000000000000F851D6CC68FC9537

→ 500 - Internal Server Error

	BLOCK 1 of 1							
	1	2	3	4	5	6	7	8
Encrypted Input	0xF9	0x51	0xD6	0xCC	0x68	0xFC	0x95	0x37
	↓	↓	↓	↓	↓	↓	↓	↓
	TRIPLE DES							
	↓	↓	↓	↓	↓	↓	↓	↓
Intermediary Value	0x39	0x73	0x23	0x22	0x07	0x6a	0x26	0x3D
	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
Initialization Vector	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
	↓	↓	↓	↓	↓	↓	↓	↓
Decrypted Value	0x39	0x73	0x23	0x22	0x07	0x6a	0x26	0x3D

INVALID PADDING

Инкрементируем последний байт IV до 0xFF, однажды он даст нам верное выравнивание (один последний байт расшифрованного текста равен единице)

Block 1 of 1	1	2	3	4	5	6	7	8
Encrypted Input	0xF8	0x51	0xD6	0xCC	0x68	0xFC	0x95	0x37
	↓	↓	↓	↓	↓	↓	↓	↓
	TRIPLE DES							
	↓	↓	↓	↓	↓	↓	↓	↓
Intermediary Value	0x39	0x73	0x23	0x22	0x07	0x6a	0x26	0x3D
	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
Initialization Vector	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x3C
	↓	↓	↓	↓	↓	↓	↓	↓
Decrypted Value	0x39	0x73	0x23	0x22	0x07	0x6a	0x26	0x01

VALID PADDING

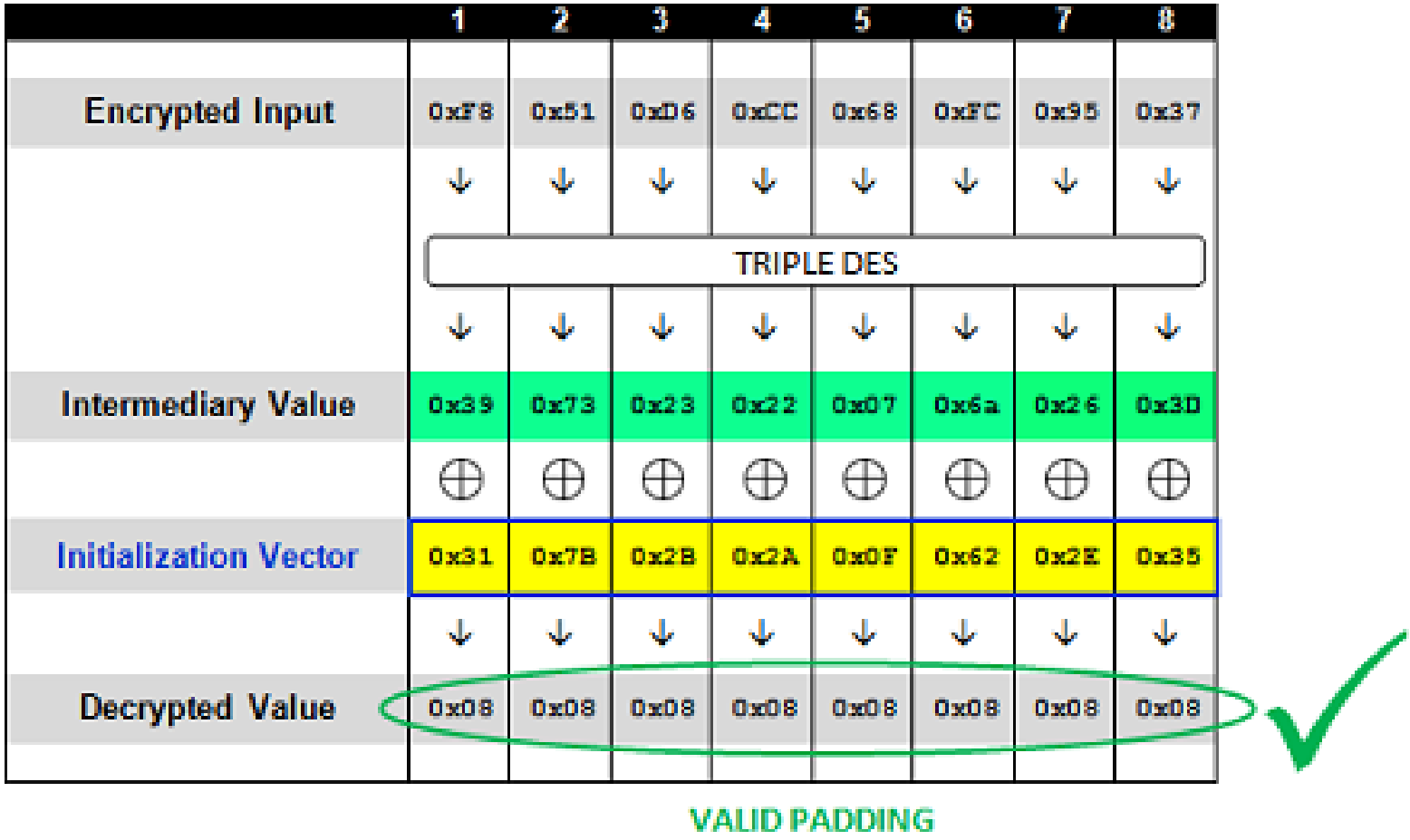
Как только мы получим корректно сформированную криптограмму (где в расшифровке выравнивание верно), ответ сервера меняется с 500 на 200 (OK).
Такие атаки называют Padding Oracle Attack, потому что сервер отвечает на вопрос, правильно ли выравнивание.

Совершенно аналогично (зная уже точно последний байт) подберём предпоследний.

	1	2	3	4	5	6	7	8
Encrypted Input	0xF8	0x51	0xD6	0xCC	0x68	0xFC	0x95	0x37
	↓	↓	↓	↓	↓	↓	↓	↓
	TRIPLE DES							
	↓	↓	↓	↓	↓	↓	↓	↓
Intermediary Value	0x39	0x73	0x23	0x22	0x07	0x6a	0x26	0x3D
	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
Initialization Vector	0x00	0x00	0x00	0x00	0x00	0x00	0x24	0x3F
	↓	↓	↓	↓	↓	↓	↓	↓
Decrypted Value	0x39	0x73	0x23	0x22	0x07	0x26	0x02	0x02

VALID PADDING

Подбираем еще шесть байт, финальная итерация будет такой:



Мы подобрали первые 8 байт Intermediary value, потратив в среднем 128 попыток на каждый байт.

	BLOCK 1 of 2								BLOCK 2 of 2							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Initialization Vector	0x7B	0x21	0x6A	0x63	0x49	0x51	0x17	0x0F	0xF8	0x51	0xD6	0xCC	0x68	0xFC	0x95	0x37
	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕	⊕
Plain-Text (Padded)	B	R	I	A	N	;	1	2	;	1	;	0x05	0x05	0x05	0x05	0x05
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Intermediary Value (HEX)	0x39	0x73	0x23	0x22	0x07	0x6A	0x26	0x3D	0xC3	0x60	0xED	0xC9	0x6D	0xF9	0x90	0x32
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
	TRIPLE DES								TRIPLE DES							
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Encrypted Output (HEX)	0xF8	0x51	0xD6	0xCC	0x68	0xFC	0x95	0x37	0x85	0x87	0x95	0xA2	0x8E	0xD4	0xAA	0xC6

Сложив это значение и исходный IV по модулю 2 получим исходные 8 байт криптограммы.

Повторяем всю процедуру со вторым 8-ми байтным блоком криптограммы.

План

CAPTCHA
HTTP — stateless protocol; состояние хранится на клиенте в виде зашифрованной строки.
В противном случае был бы лёгкий способ организовать DDoS атаки (много незакрытых подключений)

План

Так как Chosen Ciphertext Security == Non-Malleability, нам достаточно защитить сообщения от изменений злоумышленником.

Используются Message Authentication Code (MAC), гарантирующие, что сообщение не модифицировано.

Chosen Plaintext Security + использование MAC = Chosen Ciphertext Security

Так как Chosen Ciphertext Security == Non-Malleability, нам достаточно защитить сообщения от изменений злоумышленником.
Используются Message Authentication Code (MAC), гарантирующие, что сообщение не модифицировано.
Chosen Plaintext Security + использование MAC = Chosen Ciphertext Security

Возможные комбинации

|| — конкатенация

Encrypt-Then-MAC

C=E(M), t = MAC(C)

Посылаем C||t

MAC-Then-Encrypt

t = MAC(M), C = E(M||t)

Посылаем C

Encrypt-And-MAC — небезопасная

C = E(M), t = MAC(M)

Посылаем C||t

Возможные комбинации

|| — конкатенация

Encrypt-Then-MAC

C=E(M), t = MAC(C)

Посылаем C||t

MAC-Then-Encrypt

t = MAC(M), C = E(M||t)

Посылаем C

Encrypt-And-MAC — небезопасная

C = E(M), t = MAC(M)

Посылаем C||t

План

Encryption is not authentication

- Если вы шифруете сообщения, это не защищает их от модификации.
- Всегда необходима проверка целостности (Integrity check).
- Библиотечный примитив для защиты — authenticated encryption.

Материалы I


Jonathan Katz, Yehuda Lindell
Introduction to Modern Cryptography: Principles and Protocols
Chapman & Hall, 2007.


D. Bleichenbacher
Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standart PKCS, 1998.


S. Vaudenay
Security Flaws Induced by CBC Padding, 2002.


J. Rizzo, T. Duong
Practical Padding Oracle Attacks, 2010.


V. Shoup
Why chosen ciphertext security matters , IBM Research Report RZ 3076, November, 1998

Материалы II


Brian Holyfield (Gotham Digital Security Blog)
Automated Padding Oracle Attacks with PadBuster, 2010
http://blog.gdssytecurity.com/l/b/2010/09/14/automated-padding-oracle-attacks-with-padbuster/

Благодарности I

- Илья Миронов (Microsoft Research)
- Александр Куликов (АУ РАН, Computer Science Center)