конференция
РусКрипто

Ежегодная международная научно-практическая конференция

«РусКрипто'2023»

О свойстве безопасности RUP для схем аутентифицированного шифрования

Бабуева А. А., ведущий инженер-аналитик, КриптоПро
Алексеев Е. К., к.ф.-м.н., начальник отдела криптографических исследований, КриптоПро
Ахметзянова Л. Р., зам. начальника отдела криптографических исследований, КриптоПро
Божко А. А., инженер-аналитик, КриптоПро

Схемы аутентифицированного шифрования

$Enc(K, P) \rightarrow (IV, C)$: алгоритм аутентифицированного шифрования

$Dec(K, IV, C) \rightarrow P$ или $\perp$: детерминированный алгоритм расшифрования с проверкой целостности

K – ключ

IV – вектор инициализации (случайный или уникальный)

P – открытый текст

C – шифртекст

Стандартные свойства безопасности

конфиденциальность

целостность

при атаке с выбором шифртекста

На практике

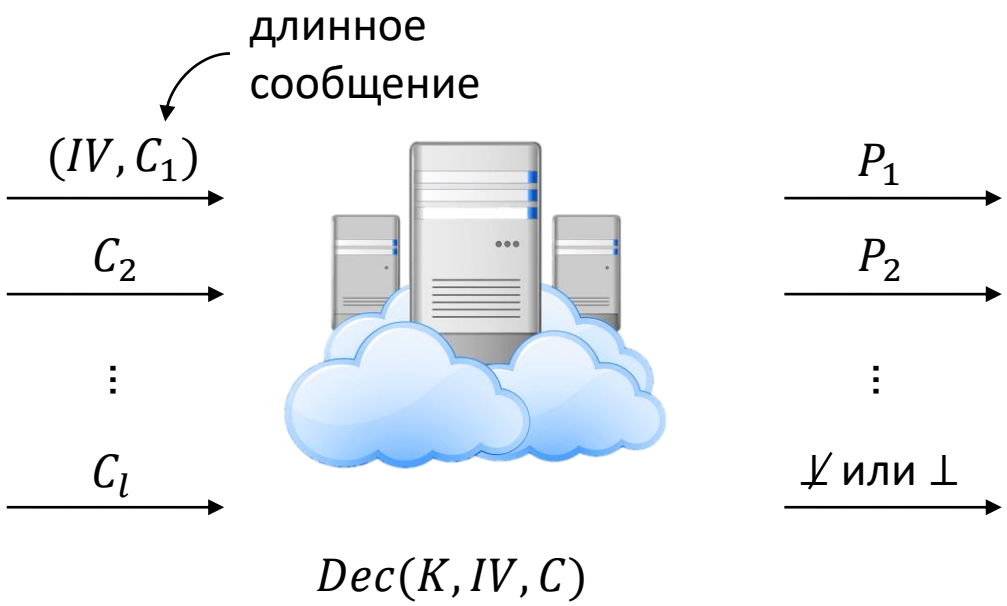
прикладная система

(IV, C)

P или $\perp$

$Dec(K, IV, C)$

На практике



поточная обработка

прикладная система

прикладная система получает результат расшифрования даже некорректных шифртекстов

⇩

он может стать доступным нарушителю

На практике

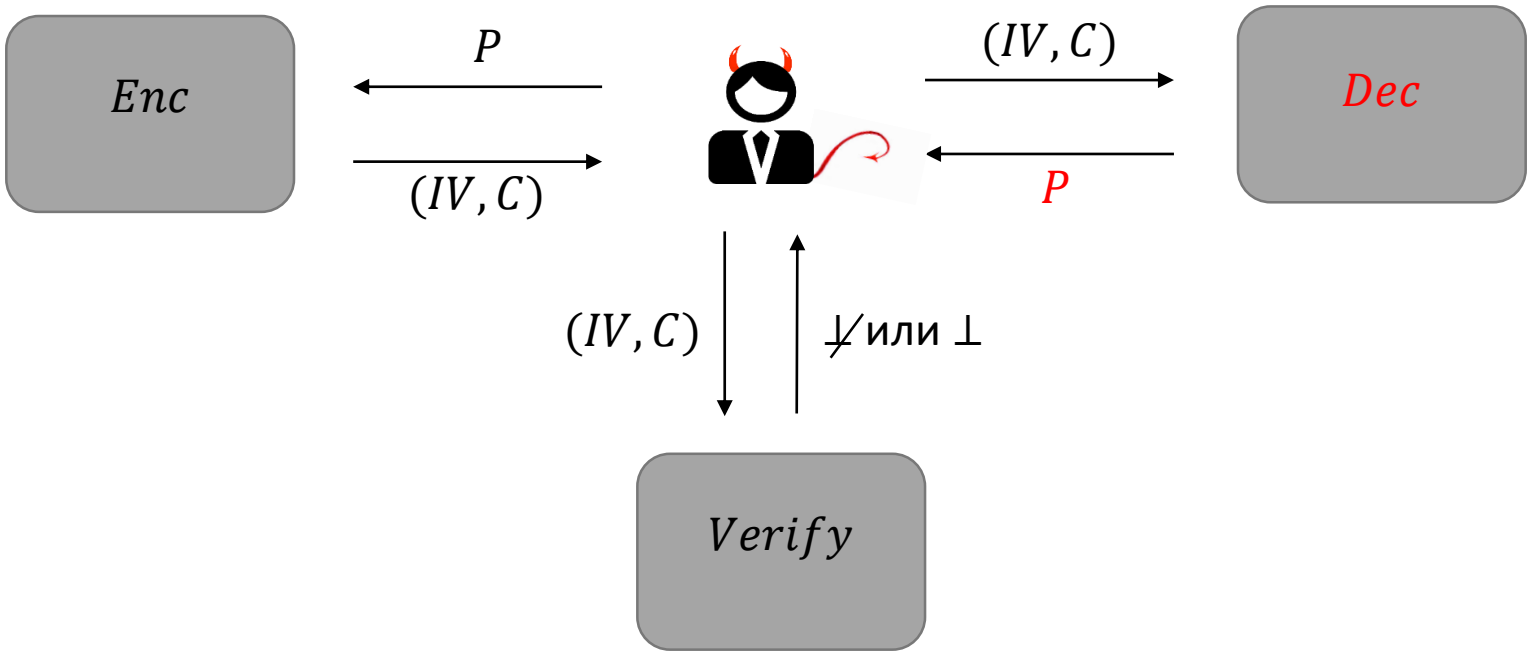
Нарушитель получает результат расшифрования некорректных шифртекстов:

конфиденциальность

целостность

Модель INT-RUP

INTegrity under Release of Unverified Plaintext



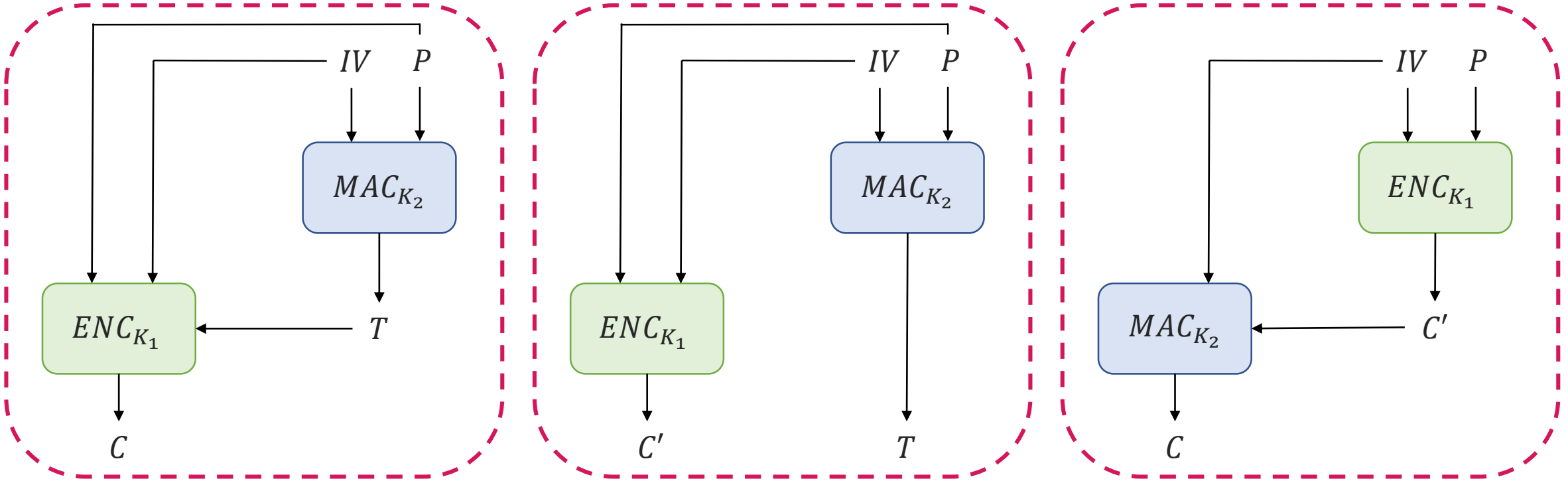
Задача нарушителя: предъявить нетривиальную подделку (IV^*, C^*)

Далее в докладе

- достаточные условия обеспечения стойкости в модели INT-RUP для классических схем аутентифицированного шифрования на основе режима гаммирования
- анализ стойкости некоторых стандартизированных механизмов аутентифицированного шифрования

Целевые схемы

шифрование в режиме гаммирования: $ENC_K(IV, P) = \Gamma(K, IV) \oplus P$



Mac-then-Encrypt (MtE)

Mac-and-Encrypt (M&E)

Encrypt-then-Mac (EtM)

Свойство MRAE-int

Misuse-Resistant Authenticated Encryption integrity

IV может повторяться

IV может повторяться

Задача нарушителя: предъявить нетривиальную подделку (IV^*, C^*)

Свойство MRAE-int: достаточное условие для INT-RUP

Misuse-Resistant Authenticated Encryption integrity

IV может повторяться

IV может повторяться

Задача нарушителя: предъявить нетривиальную подделку (IV^*, C^*)

Для $AE \in \{MtE, M\&E, EtM\}$ с ENC в режиме гаммирования:

$$MRAE\text{-}int \Rightarrow INT\text{-}RUP$$

Свойство MRAE-int: очень сильное условие

Пример (Mac-and-Encrypt):

$$T = MAC_{K=(K_2,K_3)}(IV, P) = H(K_2, P) \oplus F(K_3, IV)$$

$$C' = ENC_{K_1}(IV, P) = \Gamma(K_1, IV) \oplus P$$

Схема MAC Вегмана-Картера:

H – универсальная хэш-функция, F – псевдослучайная функция

не является MRAE-int стойким

потенциально является INT-RUP стойким

Свойство MRAE-int: очень сильное условие

Пример (Mac-and-Encrypt):

$$T = MAC_{K=(K_2,K_3)}(IV, P) = H(K_2, P) \oplus F(K_3, IV)$$

$$C' = ENC_{K_1}(IV, P) = \Gamma(K_1, IV) \oplus P$$

Атака в модели MRAE-int:

1) $C_1' \| T_1 \leftarrow Enc(IV_1, P_1)$ // $T_1 = H(K_2, P_1) \oplus F(K_3, IV_1)$
2) $C_2' \| T_2 \leftarrow Enc(IV_2, P_1)$ // $T_2 = H(K_2, P_1) \oplus F(K_3, IV_2)$
3) $C_3' \| T_3 \leftarrow Enc(IV_1, P_2)$ // $T_3 = H(K_2, P_2) \oplus F(K_3, IV_1)$
4) $T_4 \leftarrow T_1 \oplus T_2 \oplus T_3$ // $T_4 = H(K_2, P_2) \oplus F(K_3, IV_2)$
5) $C_4' \leftarrow P_2 \oplus (P_1 \oplus C_2')$
6) Вернуть подделку $(IV_2, C_4' \| T_4)$

повтор IV при шифровании

Свойство MRAE-int: очень сильное условие

Пример (Mac-and-Encrypt):

$$T = MAC_{K=(K_2,K_3)}(IV, P) = H(K_2, P) \oplus F(K_3, IV)$$

$$C' = ENC_{K_1}(IV, P) = \Gamma(K_1, IV) \oplus P$$

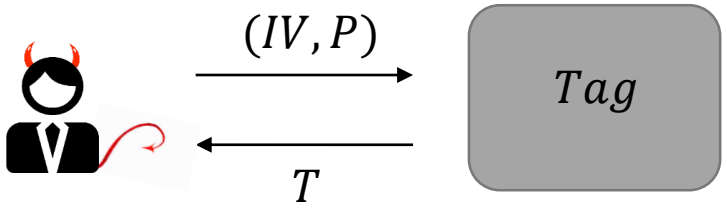
схема MAC не является стойкой при повторе IV

Атака в модели MRAE-int:

1) $C_1' \| T_1 \leftarrow Enc(IV_1, P_1)$ // $T_1 = H(K_2, P_1) \oplus F(K_3, IV_1)$
2) $C_2' \| T_2 \leftarrow Enc(IV_2, P_1)$ // $T_2 = H(K_2, P_1) \oplus F(K_3, IV_2)$
3) $C_3' \| T_3 \leftarrow Enc(IV_1, P_2)$ // $T_3 = H(K_2, P_2) \oplus F(K_3, IV_1)$
4) $T_4 \leftarrow T_1 \oplus T_2 \oplus T_3$ // $T_4 = H(K_2, P_2) \oplus F(K_3, IV_2)$
5) $C_4' \leftarrow P_2 \oplus (P_1 \oplus C_2')$
6) Вернуть подделку $(IV_2, C_4' \| T_4)$

Ослабляем условия…

Модель $nUF\text{-}CMA$ (nonce-based UnForgeability under Chosen Message Attack) для схемы MAC



IV должен быть уникальным

Задача нарушителя: предъявить нетривиальную подделку (IV^*, P^*, T^*)

Для $AE \in \{MtE, M\&E, EtM\}$ с ENC в режиме гаммирования:

$MAC - nUF\text{-}CMA$ стойкий
независимые ключи для ENC и MAC

$\Rightarrow$ $INT\text{-}RUP$

Как следствие…

Пример (Mac-and-Encrypt):

$$T = MAC_{K=(K_2,K_3)}(IV, P) = H(K_2, P) \oplus F(K_3, IV)$$

$$C' = ENC_{K_1}(IV, P) = \Gamma(K_1, IV) \oplus P$$

является INT-RUP стойким

Для $AE \in \{MtE, M\&E, EtM\}$ с ENC в режиме гаммирования :

$$INT\text{-}RUP \not\Rightarrow MRAE\text{-}int$$

А что со схемами ГОСТ?

MGM

Encrypt-then-Mac

потенциально стойкий в INT-RUP,
т.к. потенциально стойкий в MRAE-int,
см. [1]

CMS
Encrypted/EnvelopedData

Mac-then-Encrypt

построена атака в INT-RUP

[1] A. Kurochkin, D. Fomin «MGM Beyond the Birthday Bound», CTCrypt'19

Шифрование в CMS

$CMS.Enc(K = (K_{enc}, K_{mac}), P)$:

1. $IV \leftarrow_U \{0,1\}^{n/2}$
2. $T \leftarrow OMAC(K_{mac}, P)$
3. $C \leftarrow CTR\text{-}ACPKM(l, K_{enc}, IV, P \| T)$
4. return (IV, C)

$CMS.Dec(K = (K_{enc}, K_{mac}), IV, C)$:

1. $P \| T \leftarrow CTR\text{-}ACPKM(l, K_{enc}, IV, C)$
2. $T' \leftarrow OMAC(K_{mac}, P)$
3. if $(T' = T)$: return P
4. return $false$

Шифрование в CMS (в условиях RUP)

$CMS.Enc(K = (K_{enc}, K_{mac}), P)$:

1. $IV \leftarrow_U \{0,1\}^{n/2}$
2. $T \leftarrow OMAC(K_{mac}, P)$
3. $C \leftarrow CTR\text{-}ACPKM(l, K_{enc}, IV, P\|T)$
4. return (IV, C)

$CMS.Dec(K = (K_{enc}, K_{mac}), IV, C)$:

1. $P\|T \leftarrow CTR\text{-}ACPKM(l, K_{enc}, IV, C)$
2. return P

$CMS.Verify(K = (K_{enc}, K_{mac}), IV, C)$:

1. $P\|T \leftarrow CTR\text{-}ACPKM(l, K_{enc}, IV, C)$
2. $T' \leftarrow OMAC(K_{mac}, P)$
3. if $(T' = T)$: return $true$
4. return $false$

Атака на CMS

1. Делает запрос на шифрование $Enc(P) \rightarrow (IV, C)$
2. Делает запрос на расшифрование $Dec(IV, C_1) \rightarrow P_1$ и вычисляет
$$\Gamma = C_1 \oplus P_1$$
3. Делает запрос на расшифрование $Dec(IV^*, C_2) \rightarrow P_2$ и вычисляет
$$\Gamma^* = C_2 \oplus P_2$$
4. Вычисляет $C^* = \Gamma \oplus C \oplus \Gamma^*$
5. Предъявляет (IV^*, C^*) в качестве подделки

Атака на CMS

1. Делает запрос на шифрование $Enc(P) \to (IV, C)$
2. Делает запрос на расшифрование $Dec(IV, C_1) \to P_1$ и вычисляет $\Gamma = C_1 \oplus P_1$
3. Делает запрос на расшифрование $Dec(IV^*, C_2) \to P_2$ и вычисляет $\Gamma^* = C_2 \oplus P_2$
4. Вычисляет $C^* = \Gamma \oplus C \oplus \Gamma^*$
5. Предъявляет (IV^*, C^*) в качестве подделки

$C \oplus \Gamma = P \parallel T$

$C^* = (P \parallel T) \oplus \Gamma^*$

Проблема: можно «перешифровать» $(P \parallel T)$ с помощью другого IV, имитовставка не зависит от IV

Пример прикладной системы

(sk_C, pk_C)

$C = Enc_{pk_S}(tx, Sign_{sk_C}(tx))$

(sk_S, pk_S)

сервер исполнения транзакций

1) $tx, \sigma \leftarrow Dec_{sk_S}(C)$
2) проверка подписи σ
3) проверка tx на уникальность по $hash(C)$
4) выполнение транзакции tx

защита от атак повтора

Пример прикладной системы

(sk_C, pk_C)

$C = Enc_{pk_S}(tx, Sign_{sk_C}(tx))$

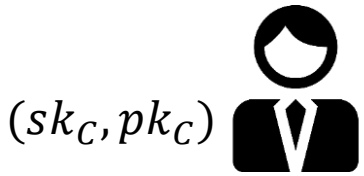
1. tx = «перевести 100 рублей клиенту B»

(sk_S, pk_S)

сервер исполнения транзакций

1) $tx, \sigma \leftarrow Dec_{sk_S}(C)$
2) проверка подписи σ
3) проверка tx на уникальность по $hash(C)$
4) выполнение транзакции tx

Пример прикладной системы



(sk_C, pk_C)

$C = Enc_{pk_S}(tx, Sign_{sk_C}(tx))$

1. tx = «перевести 100 рублей клиенту B»

(sk_S, pk_S)

сервер исполнения транзакций

2-3. навязывает длинные некорректные сообщения и получает результаты расшифрования

1) $tx, \sigma \leftarrow Dec_{sk_S}(C)$
2) проверка подписи σ
3) проверка tx на уникальность по $hash(C)$
4) выполнение транзакции tx

Пример прикладной системы

(sk_C, pk_C)

$C = Enc_{pk_S}(tx, Sign_{sk_C}(tx))$

1. tx = «перевести 100 рублей клиенту B»

(sk_S, pk_S)

сервер исполнения транзакций

проверка успешная, т.к. $C \neq C^*$

1) $tx, \sigma \leftarrow Dec_{sk_S}(C)$
2) проверка подписи σ
3) проверка tx на уникальность по $hash(C)$
4) выполнение транзакции tx

4. формирует C^*, корректный для tx

Итог: транзакция tx выполняется дважды

Что делать?

Решение. *Организационно-технические меры*

Требования по использованию механизма в случае поточной обработки данных:

1) получаемые в процессе расшифрования открытые данные не должны обрабатываться или храниться в месте, доступном нарушителю, до момента окончания процесса обработки

2) в случае завершения процесса обработки с ошибкой необходимо удалить все полученные открытые данные безопасным образом

В будущем – новая версия CMS

1) на основе MGM

2) на основе текущего стандарта

$CMS.Enc(K = (K_{enc}, K_{mac}), P)$:

1. $IV \leftarrow_U \{0,1\}^{n/2}$
2. $T \leftarrow OMAC(K_{mac}, IV \| P)$
3. $C \leftarrow CTR\text{-}ACPKM(l, K_{enc}, IV, P \| T)$
4. return (IV, C)

$CMS.Dec(K = (K_{enc}, K_{mac}), IV, C)$:

1. $P \| T \leftarrow CTR\text{-}ACPKM(l, K_{enc}, IV, C)$
2. $T' \leftarrow OMAC(K_{mac}, IV \| P)$
3. if $(T' = T)$: return P
4. return $false$

стойкость в INT-RUP обеспечивается за счет неподделываемости $OMAC$ в модели $nUF\text{-}CMA$

Спасибо за внимание!

Контактная информация:

babueva@cryptopro.ru