конференция
РусКрипто

Ежегодная международная научно-практическая конференция

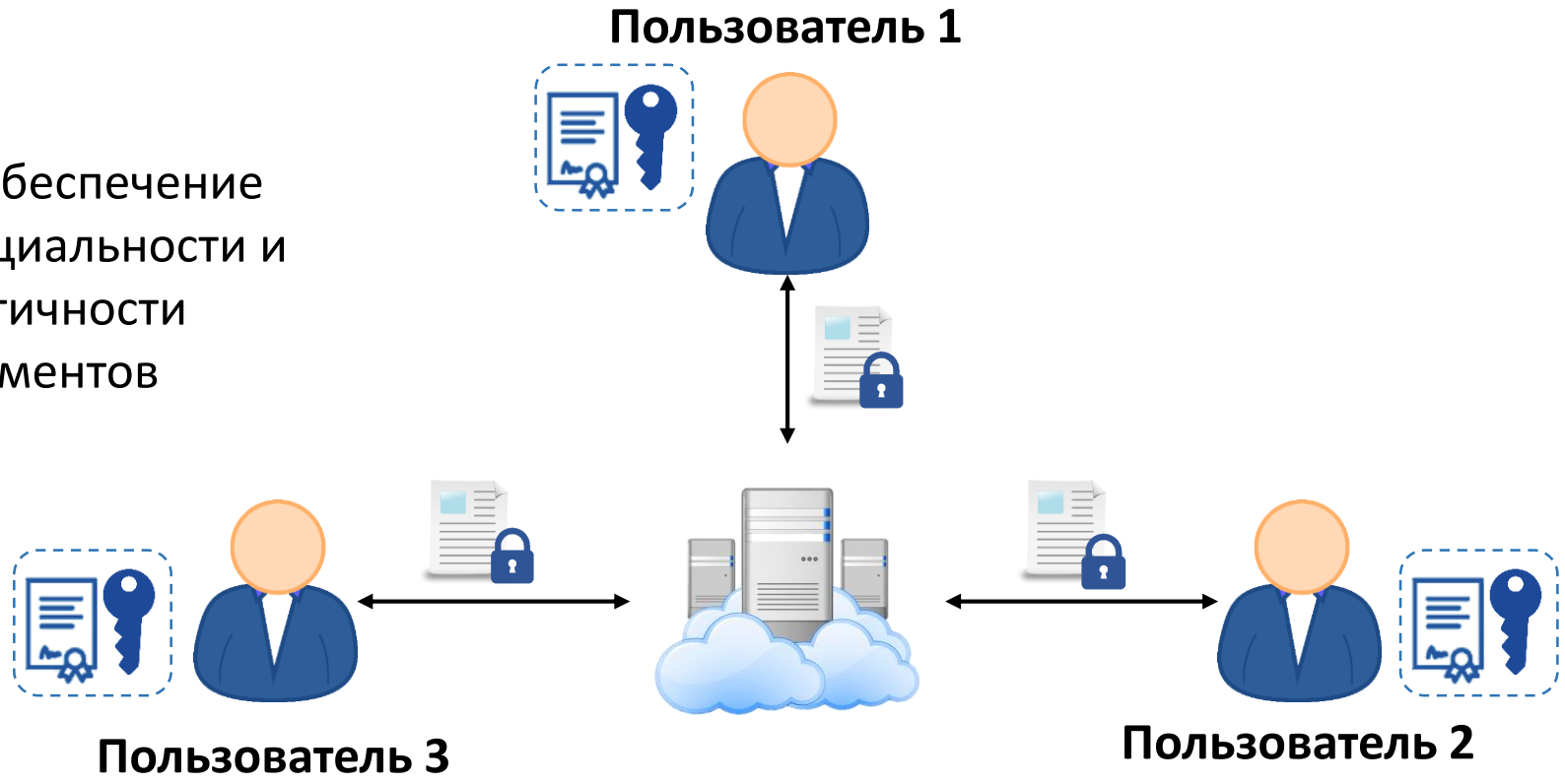
«РусКрипто'2024»

Encrypt-then-Sign или Sign-then-Encrypt, вот в чем вопрос...

Ахметзянова Л. Р., к.ф.-м.н., зам. начальника отдела криптографических исследований, КриптоПро
Алексеев Е. К., к.ф.-м.н., начальник отдела криптографических исследований, КриптоПро

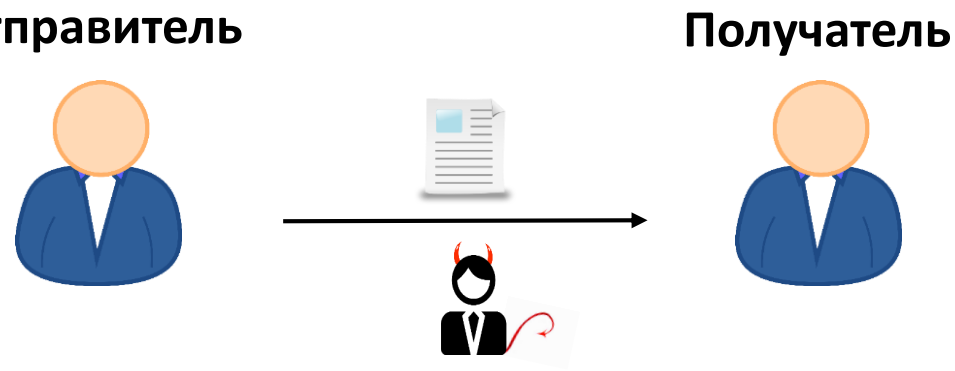
Системы электронного документооборота

Задача: обеспечение конфиденциальности и аутентичности документов



Стандартные свойства

Что значит «обеспечение конфиденциальности и аутентичности документов» в многопользовательских системах?



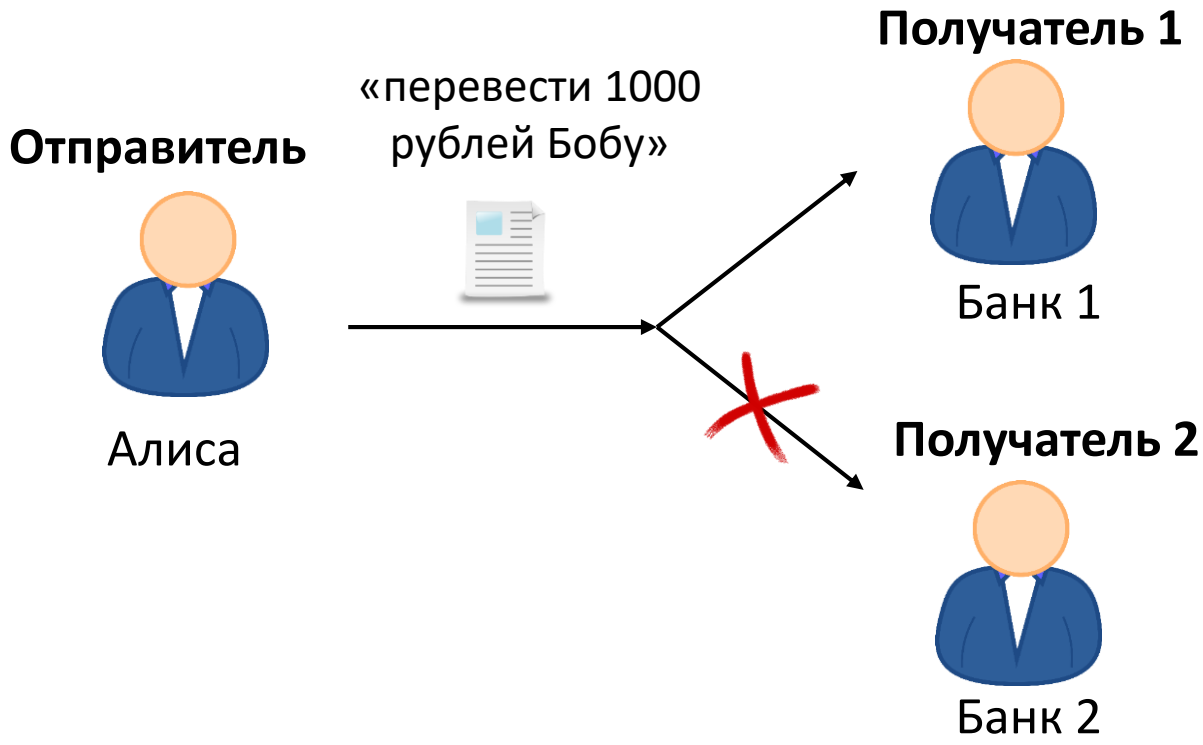
Конфиденциальность:
Никто кроме отправителя и получателя не может получить доступ к документу

Аутентичность:
Никто не может нарушить подлинность документа отправителя (в том числе, получатель)

Дополнительные свойства

Уникальность получателя
Другой получатель не может принять документ от лица отправителя, если тот не отправлял ему данный документ

Системы, для которых может быть актуально:
получатели являются банками, обрабатывающими транзакции пользователей. Пользователи, отправляющие транзакции, могут обслуживаться в нескольких банках.



Немного формализма: модель нарушителя

Активный нарушитель (атака с выбором открытого текста и шифртекста)

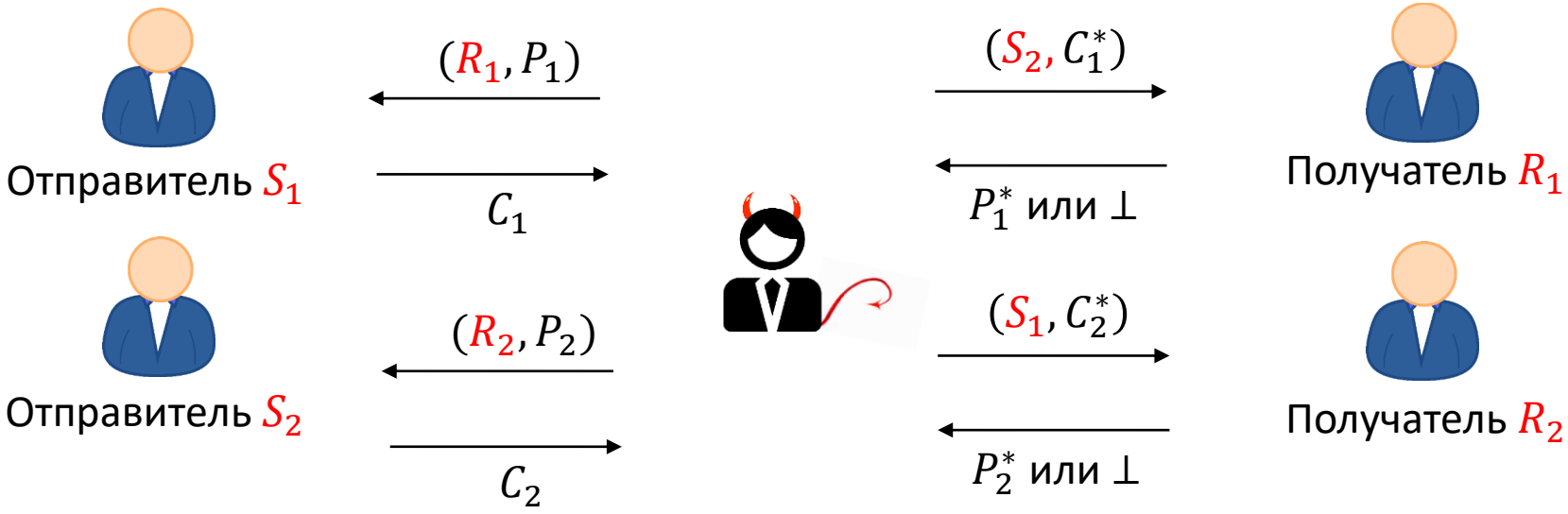
Один отправитель и один получатель:

P – открытый документ
C – защищенный документ

Немного формализма: модель нарушителя

Активный нарушитель (атака с выбором открытого текста и шифртекста)

Несколько отправителей и несколько получателей:



$C^* \neq C$ для каждой пары (S_i, R_i)

\+ нарушитель может обладать долговременными ключами участников, отличных от атакуемых

Немного формализма: модель угрозы

Конфиденциальность

Модель угрозы: получение какой-либо нетривиальной информации о секретном документе P честного отправителя S для честного получателя R

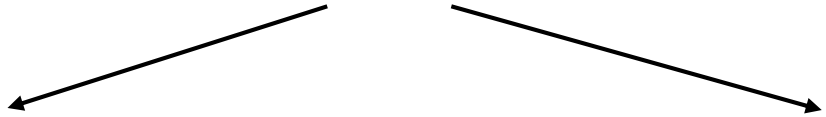
P.S. нарушитель может обладать закрытым ключом отправителя S (PFS)

Немного формализма: модель угрозы

Аутентичность

Модель угрозы: формирование такого защищенного документа C^* от лица честного отправителя S, что

- он успешно обработается на стороне некоторого получателя R и
- будет получен открытый документ P^*, который ранее не отправлялся от лица S^* в сторону …



Слабая

какого-либо получателя

Сильная
(+ уникальность получателя)

получателя R

P.S. нарушитель может обладать закрытым ключом получателя R

Как обеспечить защиту?

Комбинации схем шифрования с открытым ключом и схем подписи

Sign-then-Encrypt

Encrypt-then-Sign

Encrypt-and-Sign

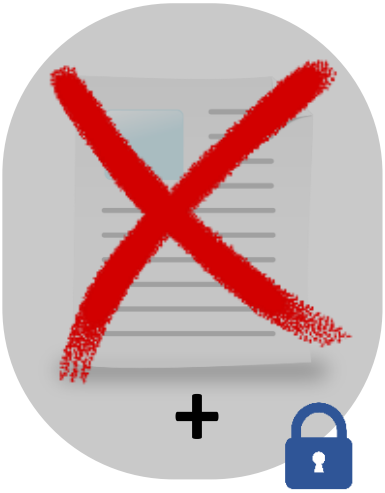
Как обеспечить защиту?

Комбинации схем шифрования с открытым ключом и схем подписи

Sign-then-Encrypt

Encrypt-then-Sign

Encrypt-and-Sign



SIGNATURE

Не обеспечивается конфиденциальность

Базовые механизмы

Схема шифрования с открытым ключом (PKE)

$Enc(pk_e, p) \rightarrow c$: алгоритм шифрования

$Dec(sk_e, c) \rightarrow p$ или $\perp$: алгоритм расшифрования

sk_e – ключ расшифрования
pk_e – ключ шифрования
p – открытый текст
c – шифртекст

Схема подписи (Sig)

$Sign(sk_s, m) \rightarrow \sigma$: алгоритм создания подписи

$Verify(pk_s, m, \sigma) \rightarrow b$: алгоритм проверки подписи

sk_s – ключ подписи
pk_s – ключ проверки подписи
m – сообщение
σ – подпись

Базовые свойства безопасности

Схема шифрования с открытым ключом (PKE)

Конфиденциальность

относительно пассивного нарушителя (IND-CPA)
относительно активного нарушителя (IND-CCA)

Схема подписи (Sig)

Неподделываемость

атака с выбором сообщений (UF-CMA)

Комбинации: Encrypt-then-Sign

Открытый ключ пользователя: $pk = (pk_e, pk_s)$

Закрытый ключ пользователя: $sk = (sk_e, sk_s)$

Send$(sk^S, pk^R, P) \to C$:

1. $c = Enc(pk_e^R, P)$
2. $\sigma = Sign(sk_s^S, c)$
3. $C = c \parallel \sigma$

Receive$(pk^S, sk^R, C) \to P$ или $\perp$:

1. $c \parallel \sigma = C$
2. $b = Verify(pk_s^S, c, \sigma)$
3. $If\ b = 0$ return $\perp$
4. $P = Dec(sk_e^R, c)$

Комбинации: Sign-then-Encrypt

Открытый ключ пользователя: $pk = (pk_e, pk_s)$

Закрытый ключ пользователя: $sk = (sk_e, sk_s)$

$\underline{\boldsymbol{Send}(sk^S, pk^R, P) \rightarrow C}$:

1. $\sigma = Sign(sk_s^S, P)$
2. $C = Enc(pk_e^R, P \parallel \sigma)$

$\underline{\boldsymbol{Receive}(pk^S, sk^R, C) \rightarrow P$ или $\perp}$:

1. $P \parallel \sigma = Dec(sk_e^R, C)$
2. $If P \parallel \sigma = \perp$ return $\perp$
3. $b = Verify(pk_s^S, P, \sigma)$
4. $If\ b = 0$ return $\perp$

Результаты. Свойства безопасности

Конфиденциальность

	IND-CPA PKE	IND-CCA PKE
Encrypt-then-Sign	–	–
Sign-then-Encrypt	–	+

Аутентичность

- только слабая ±
- сильная +

	IND-CPA PKE	IND-CCA PKE
Encrypt-then-Sign	–	–
Sign-then-Encrypt	±	±

Результаты. Свойства безопасности

Конфиденциальность

	IND-CPA PKE	IND-CCA PKE
Encrypt-then-Sign	–	–
Sign-then-Encrypt	–	+

Аутентичность
- только слабая ±
- сильная +

	IND-CPA PKE	IND-CCA PKE
Encrypt-then-Sign	–	–
Sign-then-Encrypt	±	±

Атаки: Encrypt-then-Sign

$\underline{\boldsymbol{Send}(sk^S, pk^R, P) \to C:}$
1. $c = Enc(pk_e^R, P)$
2. $\sigma = Sign(sk_S^S, c)$
3. $C = c \parallel \sigma$

Угроза: нарушение конфиденциальности

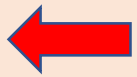
Атака для любой схемы PKE

1. Пусть честный S для секретного P формирует $C = c \parallel \sigma$ в сторону честного R.
2. Нарушитель S' **«переподписывает»** шифртекст c, формируя подпись σ' с помощью своего ключа.
3. Нарушитель отправляет сообщение $C' = c \parallel \sigma'$ от своего лица в сторону R.
3. На стороне R проверка сообщения пройдет успешно и нарушитель получит доступ к документу P.

Пример: получатель является сервисом по хранению и обработки документов; приходящие от пользователей документы загружаются в их ЛК и становятся доступны для просмотра при входе в ЛК.

Результаты. Свойства безопасности

Конфиденциальность

	IND-CPA PKE	IND-CCA PKE
Encrypt-then-Sign	–	–
Sign-then-Encrypt	– 	+

Аутентичность
- только слабая ±
- сильная +

	IND-CPA PKE	IND-CCA PKE
Encrypt-then-Sign	–	–
Sign-then-Encrypt	±	±

Атаки: Sign-then-Encrypt

$\underline{\boldsymbol{Send}(sk^S, pk^R, P) \to C:}$
1. $\sigma = Sign(sk_s^S, P)$
2. $C = Enc(pk_e^R, P \parallel \sigma)$

Угроза: нарушение конфиденциальности (для IND-CPA PKE)

Атака для определенной схемы: PKE = KEM + DEM с режимом гаммирования CTR.

1. Пусть честный S для секретного P формирует $C = \hat{C} \parallel C_P \parallel C_\sigma$ в сторону честного R.

   Пусть нарушителю известно, что $P = P_1$ либо $P = P_2$.
2. Пусть честный S для P_1 и P_3, $|P_1| = |P_3|$, формирует C_1 и C_3 в сторону нарушителя R'.
3. Нарушитель R', легитимно расшифровывая C_1 и C_2, **получает подписи** σ_1 и σ_3 для P_1 и P_3.
4. Нарушитель формирует $C'_P = C_P \oplus P_1 \oplus P_3$ и $C'_\sigma = C_\sigma \oplus \sigma_1 \oplus \sigma_3$.
5. Нарушитель отправляет сообщение $C^* = \hat{C} \parallel C_{P'} \parallel C_{\sigma'}$ от лица S в сторону честного R.
6. Если проверка на стороне R завершилась успешно, то $P = P_1$, иначе $P = P_2$.

Результаты. Свойства безопасности

Конфиденциальность

	IND-CPA PKE	IND-CCA PKE
Encrypt-then-Sign	–	–
Sign-then-Encrypt	–	+

Аутентичность

- только слабая ±
- сильная +

	IND-CPA PKE	IND-CCA PKE
Encrypt-then-Sign	–	–
Sign-then-Encrypt	±	±

Атаки: Encrypt-then-Sign

$\underline{\boldsymbol{Send}(sk^S, pk^R, P) \to C:}$

1. $c = Enc(pk_e^R, P)$
2. $\sigma = Sign(sk_s^S, c)$
3. $C = c \parallel \sigma$

Угроза: нарушение аутентичности (сильной или слабой)

Атака для определенной схемы: PKE, в котором Dec никогда не возвращает $\perp$.

1. Пусть честный S для P формирует $C = c \parallel \sigma$ в сторону честного R.
2. Нарушитель перехватывает C и отправляет его от лица S в сторону **другого** честного R'.
3. На стороне R' проверка сообщения пройдет успешно (*). При этом, если получен документ $P' \neq P$, то «ломается» слабая аутентичность, иначе сильная.

(*) Для противодействия достаточно, чтобы схема шифрования обеспечивала свойство «**робастность**»:
«the difficulty of producing a ciphertext valid under two different encryption keys».
«Robust Encryption», M. Abdalla, M. Bellare, G. Neven, TCC 2010.

Результаты. Свойства безопасности

Конфиденциальность

	IND-CPA PKE	IND-CCA PKE
Encrypt-then-Sign	–	–
Sign-then-Encrypt	–	+

Аутентичность
- только слабая ±
- сильная +

	IND-CPA PKE	IND-CCA PKE
Encrypt-then-Sign	–	–
Sign-then-Encrypt	±	±

Атаки: Sign-then-Encrypt

$\underline{\boldsymbol{Send}(sk^S, pk^R, P) \to C:}$

1. $\sigma = Sign(sk_s^S, P)$
2. $C = Enc(pk_s^R, P \parallel \sigma)$

Угроза: нарушение сильной аутентичности

Атака для любой схемы PKE

1. Пусть честный S для P формирует C в сторону нарушителя R'.
2. Нарушитель расшифровывает C, получая $P \parallel \sigma$.
3. Нарушитель «**перешифровывает**» $P \parallel \sigma$ в сторону другого честного R: $C' = Enc(pk_e^R, P \parallel \sigma)$.
4. Нарушитель отправляет сообщение C' от лица S в сторону R.
5. На стороне R проверка сообщения пройдет успешно. При этом, S не отправлял документ P в сторону R.

Просто добавь ID …

Идея: идентификатор отправителя под шифрование, идентификатор получателя под подпись.

Encrypt-then-Sign

$\underline{\boldsymbol{Send}(sk^S, pk^R, P) \to C:}$

1. $c = Enc(pk_e^R, S \parallel P)$
2. $\sigma = Sign(sk_s^S, R \parallel c)$
3. $C = c \parallel \sigma$

$\underline{\boldsymbol{Receive}(pk^S, sk^R, C) \to P \text{ или } \perp:}$

1. $c \parallel \sigma = C$
2. $b = Verify(pk_s^S, R \parallel c, \sigma)$
3. $If\ b = 0$ return $\perp$
4. $S \parallel P = Dec(sk_e^R, c)$
5. Check S ← Необходимо отдельно проверять

Sign-then-Encrypt

$\underline{\boldsymbol{Send}(sk^S, pk^R, P) \to C:}$

1. $\sigma = Sign(sk_s^S, R \parallel P)$
2. $C = Enc(pk_e^R, P \parallel \sigma)$

$\underline{\boldsymbol{Receive}(pk^S, sk^R, C) \to P \text{ или } \perp:}$

1. $P \parallel \sigma = Dec(sk_e^R, C)$
2. $IfP \parallel \sigma = \perp$ return $\perp$
3. $b = Verify(pk_s^S, R \parallel P, \sigma)$
4. $If\ b = 0$ return $\perp$

Просто добавь ID … ?

Encrypt-then-Sign

$\underline{\boldsymbol{Send}(sk^S, pk^R, P) \to C:}$

1. $c = Enc(pk_e^R, S \parallel P)$ ← для конфиденциальности
2. $\sigma = Sign(sk_s^S, R \parallel c)$ ← для слабой и сильной аутентичности
3. $C = c \parallel \sigma$

Sign-then-Encrypt

$\underline{\boldsymbol{Send}(sk^S, pk^R, P) \to C:}$

1. $\sigma = Sign(sk_s^S, R \parallel P)$ ← для сильной аутентичности
2. $C = Enc(pk_e^R, P \parallel \sigma)$

Результаты. Свойства безопасности

Конфиденциальность

	IND-CPA PKE + ID	IND-CCA PKE + ID
Encrypt-then-Sign	–	+
Sign-then-Encrypt	–	+

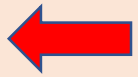
Аутентичность
- только слабая ±
- сильная +

	IND-CPA PKE + ID	IND-CCA PKE + ID
Encrypt-then-Sign	+	+
Sign-then-Encrypt	+	+

An, J.H., Dodis, Y., Rabin, T. On the Security of Joint Signature and Encryption. EUROCRYPT 2002
(получено сведение к стойкости PKE и Sig)

Результаты. Свойства безопасности

Конфиденциальность

	IND-CPA PKE + ID	IND-CCA PKE + ID
Encrypt-then-Sign	– 	+
Sign-then-Encrypt	–	+

Аутентичность
- только слабая ±
- сильная +

	IND-CPA PKE + ID	IND-CCA PKE + ID
Encrypt-then-Sign	+	+
Sign-then-Encrypt	+	+

Атаки: Encrypt-then-Sign

$\underline{\boldsymbol{Send}(sk^S, pk^R, P) \to C:}$
1. $c = Enc(pk_e^R, S \parallel P)$
2. $\sigma = Sign(sk_s^S, R \parallel c)$
3. $C = c \parallel \sigma$

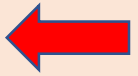
Угроза: нарушение конфиденциальности (для IND-CPA PKE)

Атака на определенную схему: PKE = KEM + DEM с режимом гаммирования CTR

1. Пусть честный S для секретного P формирует $C = c \parallel \sigma$ в сторону честного R, где $c = \hat{c} \parallel c_S \parallel c_P$.
2. Нарушитель S' **«переподписывает»** шифртекст c, формируя подпись σ' с помощью своего ключа.
3. Нарушитель формирует шифртекст $c'_S = c_S \oplus S \oplus S'$.
4. Нарушитель отправляет $C' = \hat{c} \parallel c'_S \parallel c_P \parallel \sigma'$ от лица S в сторону R.
5. На стороне R проверка пройдет успешно и нарушитель S' получит доступ к документу P.

Результаты. Свойства безопасности

Конфиденциальность

	IND-CPA PKE + ID	IND-CCA PKE + ID
Encrypt-then-Sign	–	+
Sign-then-Encrypt	– 	+

Аутентичность
- только слабая ±
- сильная +

	IND-CPA PKE + ID	IND-CCA PKE + ID
Encrypt-then-Sign	+	+
Sign-then-Encrypt	+	+

Атаки: Sign-then-Encrypt

$\underline{\boldsymbol{Send}(sk^S, pk^R, P) \to C:}$
1. $\sigma = Sign(sk_s^S, R \parallel P)$
2. $C = Enc(pk_e^R, P \parallel \sigma)$

Угроза: нарушение конфиденциальности

Атака на определенную схему: PKE = KEM + DEM с режимом гаммирования CTR

1. Пусть честный S для секретного $P = \mathrm{PIN} \parallel P'$ формирует $C = \hat{C} \parallel C_{PIN} \parallel C_{P'} \parallel C_\sigma$ в сторону честного R.

   Пусть PIN – секретное небольшое число, P' известен, $|C'_P| = |C_\sigma|$. Положим $i = 0$.
2. Нарушитель S' формирует подпись σ_i **с помощью своего ключа** для значения-догадки $\mathrm{PIN} = i$.
3. Нарушитель формирует шифртекст $C'_{\sigma_i} = C_{P'} \oplus P' \oplus \sigma_i$.
4. Нарушитель отправляет $C^* = \hat{C} \parallel C_{PIN} \parallel C'_{\sigma_i}$ от своего лица в сторону R.
5. Если проверка на стороне R завершилась успешно, то $\mathrm{PIN} = i$, иначе $i = i + 1$, перейти на шаг 2.

P.S. Атака применима и в случае добавления S под шифрование.

Выводы

Если:

1) используется PKE-схема, стойкая относительно активного нарушителя (*);
2) «замешиваются» идентификаторы отправителя и получателя;

обе схемы Sign-then-Encrypt и Encrypt-then-Sign являются потенциально стойкими в целевых моделях угроз и нарушителя.

(*) Для отечественной версии формата CMS Enveloped Data рекомендуется использовать IND-CCA-стойкие (есть имитовставка) алгоритмы с идентификаторами:

✓ id-gostr3412-2015-kuznyechik-**ctracpkm-omac**
id-gostr3412-2015-magma-**ctracpkm-omac**.

Выводы

Однако в схеме **Encrypt-then-Sign:**

- необходимо «замешивать» два идентификатора R и S **либо** замешивать только S и использовать робастную схему PKE (но нужны доп. исследования);
- после расшифрования отдельно проверять S;
- не позволяет обеспечить анонимность отправителя (например, по сообщению и подписи типа Эль-Гамаля можно «почти» однозначно восстановить открытый ключ), а при «замешивании» идентификатора R – и анонимность получателя.

Финалист

Спасибо за внимание!

Контактная информация:

lah@cryptopro.ru

alekseev@cryptopro.ru