Уязвимость ROCA и другие возможности внедрения закладок в алгоритм RSA

Маркелова А.В.

к.ф.-м.н.

НТЦ Альфа-Проект

МГТУ им. Н.Э. Баумана

конференция РусКрипто

20-23 марта 2018

2017 год: уязвимость ROCA

- отозваны 760 тысяч сертификатов ID-карт Эстонии
- прекращен выпуск смарт-карт Gemalto IDPrime.NET
- Microsoft, Google и Infineon предупредили о возможных проблемах с механизмом защиты TPM-продуктов (HP, Acer, Fujitsu, Lenovo, LG и другие)
- обнаружено 237 факторизуемых ключей RSA, используемых для подписи публикуемого на GitHub ПО
- 956 PGP-ключей RSA оказались факторизуемыми (Yubikey 4 ?)

«Ответственное шифрование»

«Такое шифрование, которое не вскрывается по предписанию суда, подрывает конституционный баланс, ставя приватность граждан выше общественной безопасности. Те зашифрованные коммуникации, которые нельзя перехватить, и те запертые криптографией устройства, которые нельзя открыть, – всё это зоны беззакония, позволяющие преступникам и террористам действовать без их выявления полицией, без обязанности отвечать перед судьями и судами присяжных»

Род Розенштейн, заместитель генерального прокурора США,

октябрь *2017*

Законодательные инициативы

Германия:

- законопроект, обязывающий производителей оборудования внедрять в свои устройства бэкдоры (все типы современных устройств, включая автомобили, телефоны, компьютеры, устройства из сферы «Интернета вещей» и пр.)

Австрия:

- законопроект, позволяющий спецслужбам тайно следить за перепиской пользователей мессенджеров WhatsApp и Skype

Россия:

- пакет Яровой

Криптографические закладки

1. Слабые закладки
2. Передача информации по скрытым каналам
3. SETUP-механизмы (Secretly Embedded Trapdoor with Universal Protection)

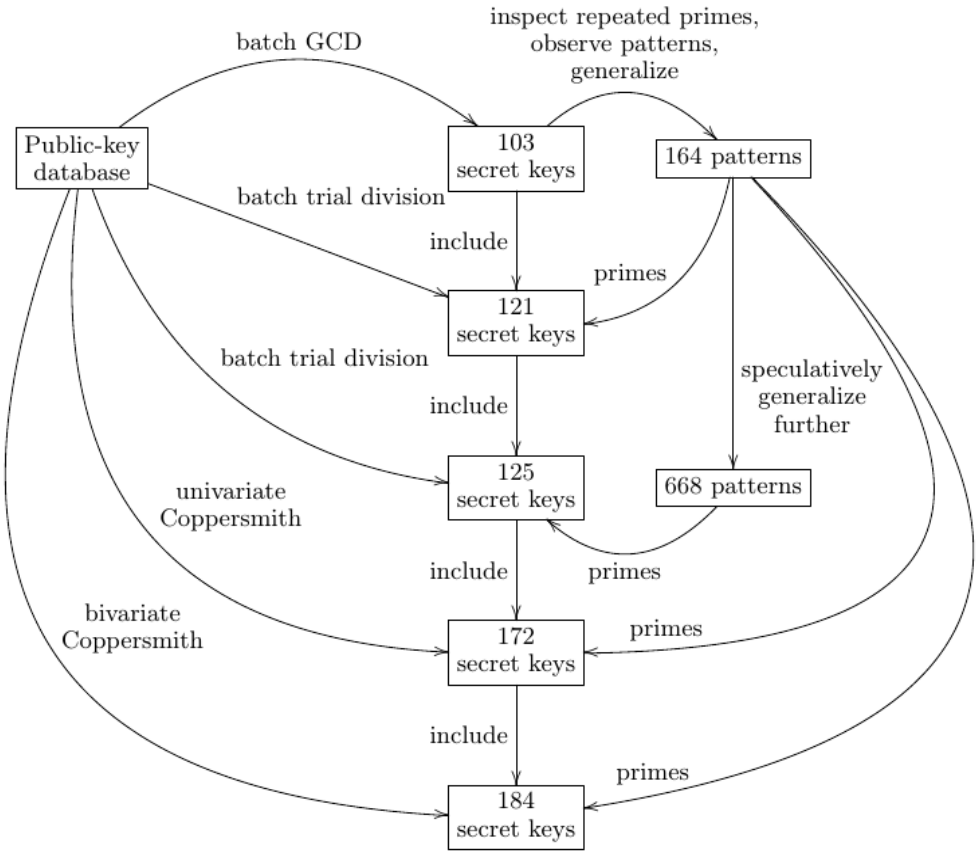
Слабые закладки

Намеренное ослабление алгоритма, позволяющее узнать секретную пользовательскую информацию на основе открытой информации.

Пример. Использование фиксированного простого числа p в генераторе ключей RSA.

Citizen Digital Certificates, Тайвань

Daniel J. Bernstein, Yun-An Chang, Chen-Mou Cheng, Li-Ping Chou, Nadia Heninger, Tanja Lange, and Nicko van Someren. 2013. Factoring RSA Keys from Certified Smart Cards: Coppersmith in the Wild. In Advances in Cryptology - ASIACRYPT 2013. Springer-Verlag, 341–360.



RENESAS

- *FIPS 140-2 Level 2*
- *Common Criteria level EAL4+*
- *Protection Profile BSI-PP-0002-2001*

RSALib, Infineon

Matus Nemec, Marek Sys, Petr Svenda, Dusan Klinec, Vashek Matyas. The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli. 2017

Ключи в RSALib имеют вид:

$$p = k*M + (65537^{a} \bmod M)$$

$$q = l*M + (65537^{b} \bmod M)$$

$$n = s*M + (65537^{a+b} \bmod M)$$

Key size	M
512 b	$P_{39}\# = 167\#$
1024 b	$P_{71}\# = 353\#$
2048 b	$P_{126}\# = 701\#$
3072 b	$P_{126}\# = 701\#$
4096 b	$P_{225}\# = 1427\#$

$M=P_m=2*3*5*\ldots*p_m$

Время взлома RSA для ROCA

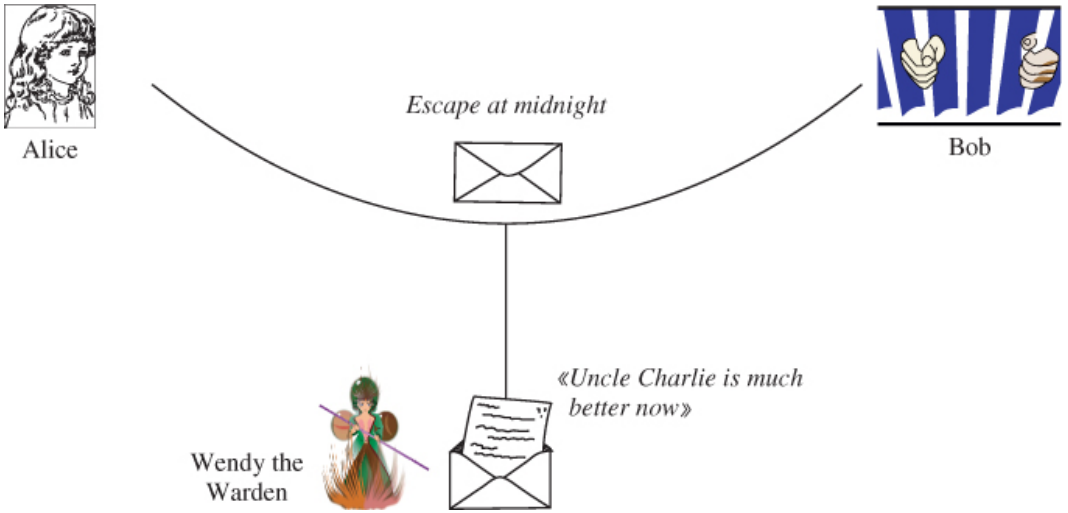
Key size	University cluster (Intel E5-2650 v3@3GHz Q2/2014)	Rented Amazon c4 instance (2x Intel E5-2666 v3@2.90GHz, estimated)
512 b	1.93 CPU hours *(verified)*	0.63 hours, $0.063
1024 b	97.1 CPU days *(verified)*	31.71 days, $76
2048 b	140.8 CPU years	45.98 years, $40,305
3072 b	$2.84 * 10^{25}$ years	$9.28 * 10^{24}$ years, $\$8.13 * 10^{27}$
4096 b	$1.28 * 10^{9}$ years	$4.18 * 10^{8}$ years, $\$3.66 * 10^{11}$

- **1000 ядер Amazon AWS:**
  - RSA-1024: 45 минут
  - RSA-2048: 17 дней

Скрытые каналы: проблема заключенного

1) **Алгоритм общения по скрытому каналу известен, но это не мешает Алисе и Бобу общаться**

2) **Вилли (Венди) не может ни прочитать информацию в скрытом канале, ни даже однозначно определить факт его использования**



1. Отправитель Алиса выбирает безвредное сообщение для подписи

2. Алиса подписывает безвредное сообщение, скрывая при этом реальное сообщение в подписи

3. Подпись и безвредное сообщение передаются Вилли

4. Вилли проверяет подпись и, не найдя в ней ничего опасного, передаёт информацию Бобу

5. Боб проверяет подпись, чтоб убедиться в аутентичности сообщения

6. Боб извлекает реальное сообщение из подписи, используя секретный ключ Алисы

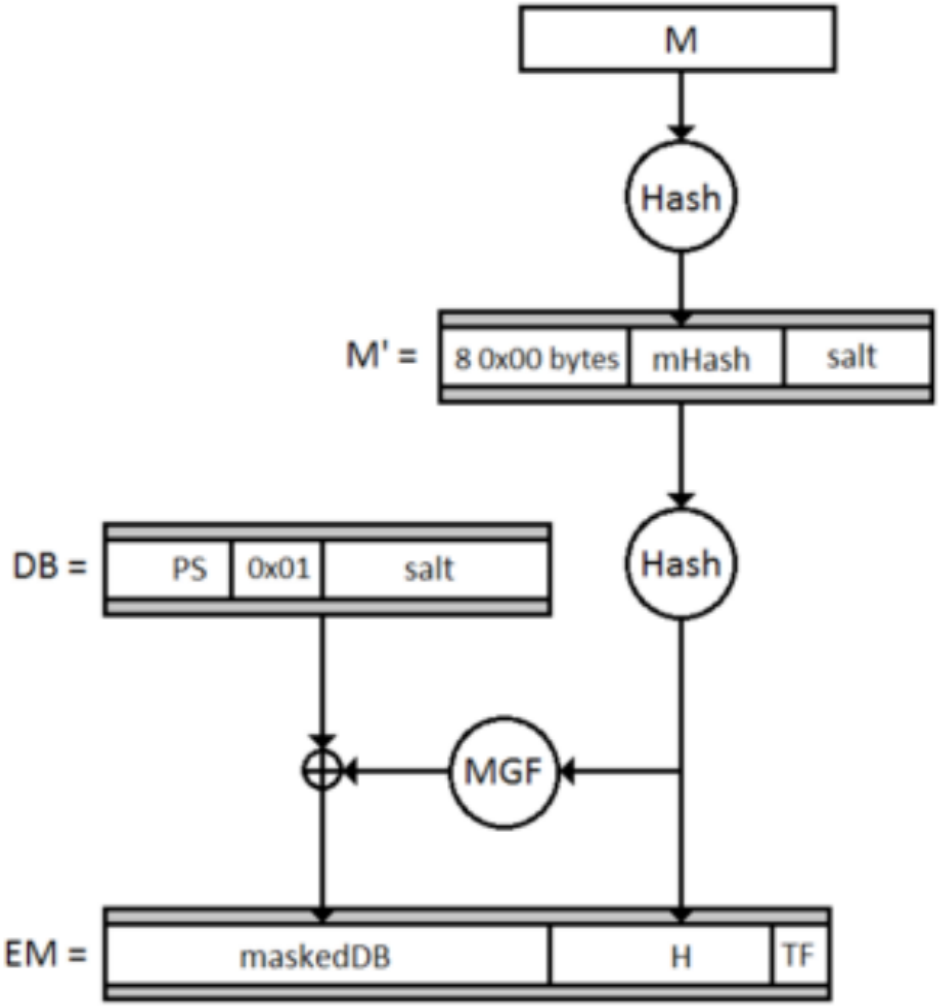
Скрытые каналы в RSASSA-PSS (PKCS#1)

E(.) – некоторый алгоритм шифрования с открытым ключом

D(.) – обратное преобразование (известно разработчику)

$m`$ – секретное сообщение

$salt`$=E($m`$)



Скрытые каналы в генераторе ключей RSA

E(.) – некоторый алгоритм шифрования с открытым ключом
D(.) – обратное преобразование (известно разработчику)
$m`$ – секретное сообщение

1. Использование E(m`) в качестве открытой экспоненты

2. Использование E(m`) в качестве старших бит открытого модуля n

SETUP-механизм

Пусть C – это криптосистема с объявленной спецификацией. SETUP-механизм – это такая модификация криптоалгоритма C в алгоритм C1, что:

1. Параметры входа C1 согласуются с объявленной спецификацией входных параметров C.

2. Параметры выхода C1 соответствуют объявленной спецификации параметров выхода C. В тоже время, выход C1 содержит секретные биты (например, биты секретного ключа пользователя), которые легко извлекаются разработчиком.

3. По выходам алгоритмы C1 и C полиномиально неразличимы для всех, кроме разработчика.

4. Выход C1 эффективно вычисляется с использованием встроенной в C1 функции шифрования с открытым ключом E, а также, возможно, других функции, содержащихся в C1.

5. Секретная функция расшифрования D, обратная к E, не содержится в C1 и известна только разработчику.

6. После обнаружения в реализации алгоритма SETUP-механизма и выяснения его особенностей, например, при помощи реверс-инженеринга программы или аппаратного защищенного устройства, и пользователи и злоумышленники (все за исключением разработчика) не могут определить использованные (или будущие) секретные ключи пользователя. В этом смысле SETUP-механизм обеспечивает «криптостойкость» системы по отношению ко всем злоумышленникам, кроме ее разработчика.

SETUP: PAP (Pretty-Awful-Privacy)

Young A., Yung M. Malicious Cryptography. Exposing Cryptovirology. Wiley Publishing, Inc. 2004

input: none
output: $W/2$-bit primes p and q such that $p \neq q$ and $|pq| = W$
1. if $i > \Theta$ then output $GenPrivatePrimes1()$ and halt
2. update i in non-volatile memory to be $i = i + 1$
3. let I be the $|\Theta|$-bit representation of i
4. for $j = 0$ to ∞ do:
5. choose x randomly from $\{0, 1, 2, ..., N - 1\}$
6. set $c_0 = x$
7. if $gcd(x, N) = 1$ then
8. choose bit b randomly and choose u randomly from $\mathbb{Z}_N^*$
9. if $J(x/N) = +1$ then set $c_0 = e_0^b e_2^{1-b} u^2 \ mod \ N$
10. if $J(x/N) = -1$ then set $c_0 = e_1^b e_3^{1-b} u^2 \ mod \ N$
11. compute $(e, c_1) = PBRM(N, 2^{W/2}, c_0)$
12. if $e = -1$ then continue
13. if $u > -u \ mod \ N$ then set $u = -u \ mod \ N$ /* for faster decr. */
14. let T_0 be the $W/2$-bit representation of u
15. for $k = 0$ to ∞ do:
16. compute $p = H(T_0||ID||I||j, \frac{kW}{2}, \frac{W}{2})$
17. if $p \geq 2^{W/2-1} + 1$ and p is prime then break
18. if $p < 2^{W/2-1} + 1$ or if p is not prime then continue
19. $c_2 = RandomBitString1()$
20. compute $n' = (c_1 \ || \ c_2)$
21. solve for the quotient q and the remainder r in $n' = pq + r$
22. if q is not a $W/2$-bit integer or if $q < 2^{W/2-1} + 1$ then continue
23. if q is not prime then continue
24. if $|pq| < W$ or if $p = q$ then continue
25. if $p > q$ then interchange the values p and q
26. set $S = (p, q)$ and break
27. output S, zeroize everything in memory except i, and halt

Условия построения лазейки с универсальной защитой

- $M = 2 * 3 * 5 * \ldots * p_k * P$, P - «большое» простое
- $\log_2 M \sim \log_2 n/4$
- $w = 1 + Pt$, $(w, p_i)=1$
- есть достаточно много простых чисел вида

$$k*M + (w^a \bmod M)$$

Уязвимость ROCA и другие возможности внедрения закладок в алгоритм RSA

Спасибо за внимание.

Вопросы?