امنیت داده ها

دکتر یعقوب فرجامی

عضو هیات علمی دانشکده فنی قم

فصل هشتم : رمزنگاری کلید عمومی
RSA

رمزگذاري كليد عمومي (Public Key Cryptography)

- در هر يك از الگوهاي رمزنگاري كه مورد بحث قرار گرفتند لازم است كه فرستنده پيام و گيرنده پيام كليد رمز را بدانند .
- وقتي فرستندة پيام از كليدي براي رمزنگاري استفاده مي‌كند و گيرندگان هم از همان كليد براي رمزگشايي بهره مي‌برند ، افشا شدن كليد رمز توسط يكي از گيرندگان پيام ، امنيت را به خطر مي‌اندازد.
- در الگوهاي جديدِ رمزگذاري، براي حل مشكل از دو كليد متفاوت استفاده مي‌شود
- يك كليد براي رمز كردن پيام و كليد ديگر براي رمزگشائي آن
- با كليدِ مخصوص رمزنگاري نمي‌توان رمزگشائي پيام را انجام داد
- بنابراين رمزكنندة پيام خودش كليدي دارد كه حتي معتمدين و گيرندگان پيام هم آنرا لازم ندارند چرا كه فقط براي رمزنگاري بكار مي‌آيد و افشا شدن آن هم لطمه اي به كسي نمي‌زند
- چرا كه با آن كليد نمي‌توان متون رمز شده را برگرداند و پيدا كردن كليد رمزگشائي از روي كليد رمزنگاري كار ساده اي نيست و هنوز امكان پذير نشده است.
-

جایگاه عملی رمزنگاری کلید عمومی

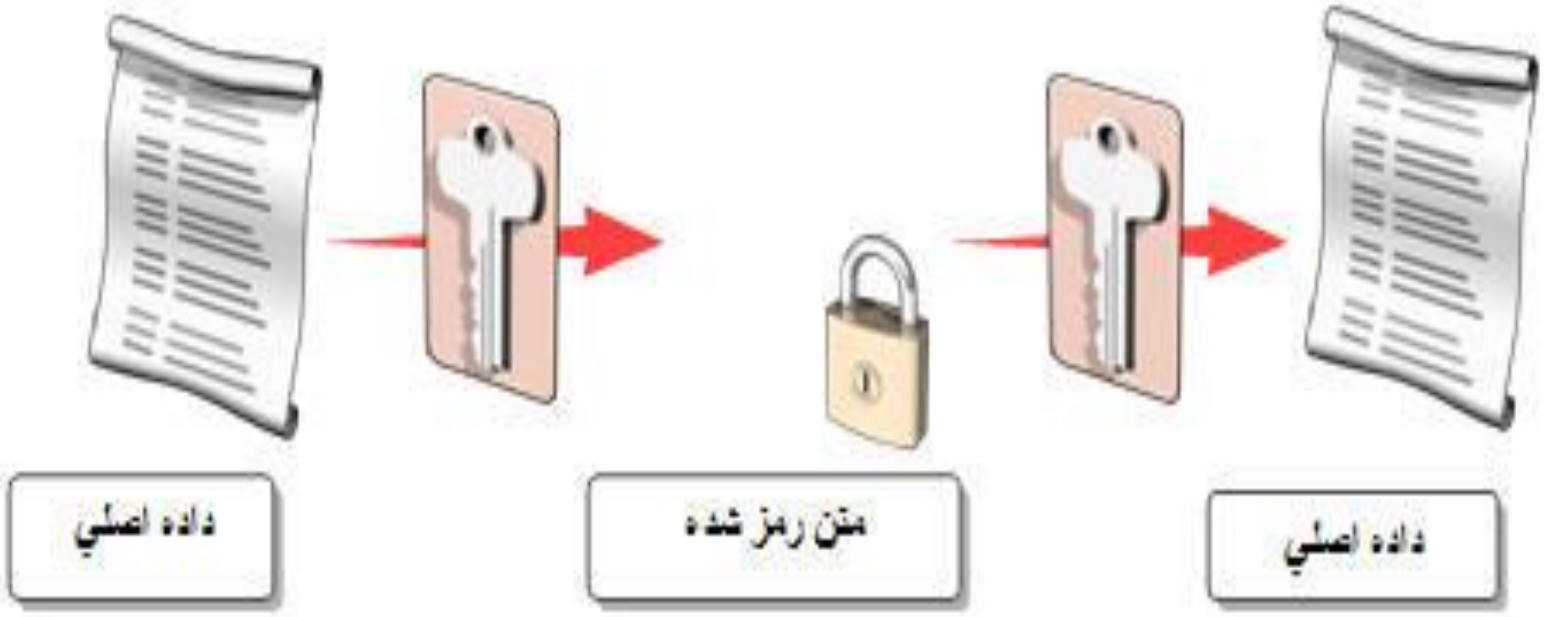
- **کلیدهای این نوع از الگوریتمها بسیار طولانی تر از الگوریتمهای مرسوم (کلید پنهان) میباشند.**
  - **الگوریتم RSA با پیمانه ۱۰۲٤ بیتی امنیتی در حد الگوریتمهای متقارن با کلیدهای ۸۰ بیتی دارد.**
- **سرعت الگوریتمهای کلید عمومی از الگوریتمهای رمزگذاری مرسوم پایین تر است.**
  - **RSA تقریباً ۱۰۰۰ بار کند تر از رمزهای کلید پنهان (با امنیت یکسان) میباشد.**

انواع روشهای رمزنگاری مبتنی بر کلید:

- الگوریتمهای کلید متقارن:
  - رمز گذاری و رمز برداری با یک کلید انجام می گیرد.

- الگوریتمهای کلید نامتقارن (کلید عمومی):
  - هر فرد یک کلید عمومی و یک کلید خصوصی دارد.
    - دفی هلمن و RSA نمونه ای از این الگوریتمها ست.
    - کاربرد در امضای دیجیتال.
    - کلیدهاي اين نوع از الگوريتمها (نامتقارن) بسيار طولاني‌تر از الگوريتمهاي مرسوم (کلید متقارن) ميباشند.

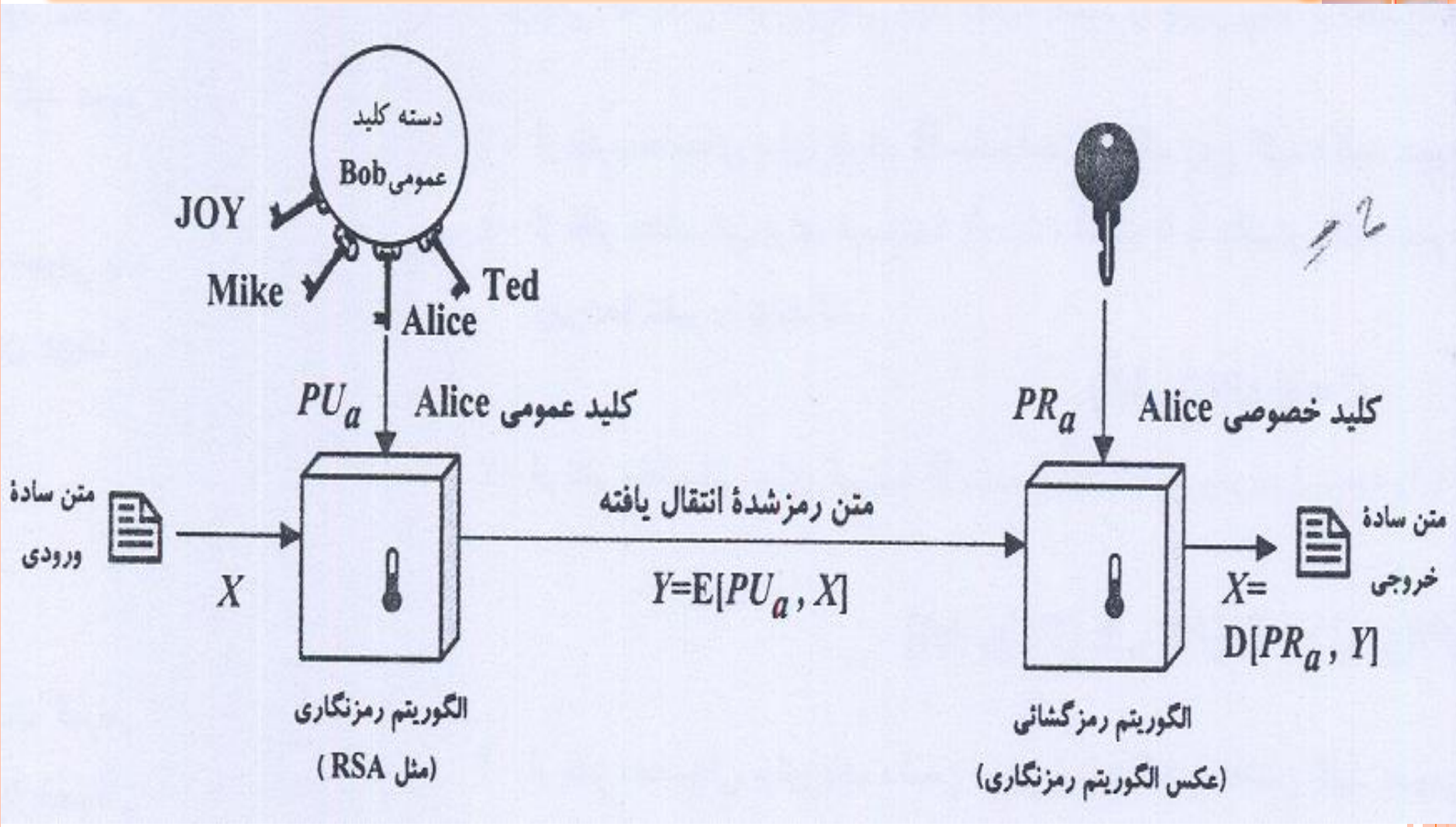
الگوریتمهای کلید متقارن:



الگوریتم‌های رمزنگاری نامتقارن

- رمزنگاری کلید عمومی در سال ۱۹۷۶ توسط Diffie و Hellman ابداع شد.
- مهمترین اختلاف رمزنگاری متقارن با رمزنگاری نامتقارن استفاده از دو کلید مجزا است.
- سوال: آیا رمزنگاری کلید عمومی از رمزنگاری سنتی امن‌تر است؟
- اجزای رمزنگاری نامتقارن:
  - متن ساده، الگوریتم رمزنگاری، کلید عمومی و خصوصی، متن رمزشده، الگوریتم رمزگشایی.
  - کلید عمومی برای همگان قابل دسترسی و شناسایی بوده و جهت رمزنگاری و کلید خصوصی تنها برای صاحب آن قابل دسترسی و شناسایی و برای رمزگشایی بکارمی‌رود.

عملیات رمزنگاری نامتقارن -رمزنگاری



عملیات رمزنگاری نامتقارن

- مراحل رمزنگاری و رمزگشایی نامتقارن: ارسال پیام توسط Bob به Alice .
  - تولید یک زوج کلید(کلید عمومی و خصوصی) توسط کاربر برای رمزنگاری و رمزگشایی.
  - هر کاربر کلید عمومی خود را در یک فایل قابل دسترس قرار می‌دهد. کاربران مجموعه‌ای از کلیدهای عمومی دیگر کاربران را در اختیار دارند.
  - Bob پیام خود را با استفاده از کلید عمومی Alice رمز کرده و ارسال می‌کند.
  - Alice پس از دریافت پیام، آن را با استفاده ازکلید خصوصی خود رمزگشایی می‌کند.
  - تا زمانی‌که از کلیدهای خصوصی افراد در شبکه بدرستی محافظت شود، ارتباطات امن خواهد بود.

الگوریتم‌های رمزنگاری نامتقارن – کاربردها

- کاربردهای رمزنگاری کلید عمومی:
  - رمزنگاری و رمزگشایی
  - امضای دیجیتال
    - فرستنده پیام را با کلید خصوصی خود امضا می‌کند.
  - توزیع کلید
    - ارسال کلیدهای متقارن
- الگوریتم‌های رمزنگاری کلید عمومی:
  - RSA، Deffie – Hellman و DSS (Al-Gamal)، ECC، ECDSA

کاربردهای سیستم‌های رمزنگاری کلید- عمومی

الگوریتم	رمزنگاری/ رمزگشائی	امضاء دیجیتال	مبادلهٔ کلید
RSA	بلی	بلی	بلی
Diffie-Hellman	خیر	خیر	بلی
DSS	خیر	بلی	خیر

رمزنگاری کلید عمومی

- کلید های رمزگذاری و رمزگشایی متفاوت اما مرتبط هستند.
- رسیدن به کلید رمز گشایی از کلید رمزگذاری از لحاظ محاسباتی ناممکن می باشد.
- رمزگذاری امری همگانی میباشد و اساساً نیازی به اشتراک گذاشتن اطلاعات محرمانه ندارد.
- رمز گشایی از طرف دیگر امری اختصاصی بوده و محرمانگی پیامها محفوظ میماند.

رمزگذاری کلید عمومی

- برای رمز نگاری کلید عمومی گامهای زیر را برمیداریم:

1. هر کاربر یک زوج کلید رمزگذاری و رمز گشایی تولید میکند.
2. کاربران کلید رمزگذاری خود را به صورت عمومی اعلان میکنند درحالی که کلید رمز گشایی مخفی میباشد.
3. همگان قادر به ارسال پیام رمز شده برای هر کاربر دلخواه با استفاده از کلید رمزگذاری (عمومی) او میباشند.
4. هر کاربر میتواند با کمک کلید رمزگشایی (خصوصی) پیامهایی که با کلید رمزگذاری (عمومی) او رمز شده رمزگشایی کند.

جایگزینی یا تکمیل؟

از نظر کاربردی، رمزگذاری با کلید عمومی بیش از آنکه جایگزینی برای رمزگذاری مرسوم باشد نقش مکمل آنرا برای حل مشکلات توزیع کلید بازی می کند.

Misconceptions!

دو تصور اشتباه دیگر درباره کلید عمومی

– رمزنگاری با کلید عمومی امن تر است!

- در هر دو روش رمزنگاری امنیت به طول کلید وابسته است

– مسئله توزیع کلید در رمزنگاری با کلید عمومی برطرف شده است

- چگونه مطمئن شویم کلید عمومی لزوما متعلق به شخص ادعاکننده است؟!
- توزیع کلید عمومی آسانتر است، ولی بدیهی نیست.

محرمانگی و احرازاصالت بطور همزمان

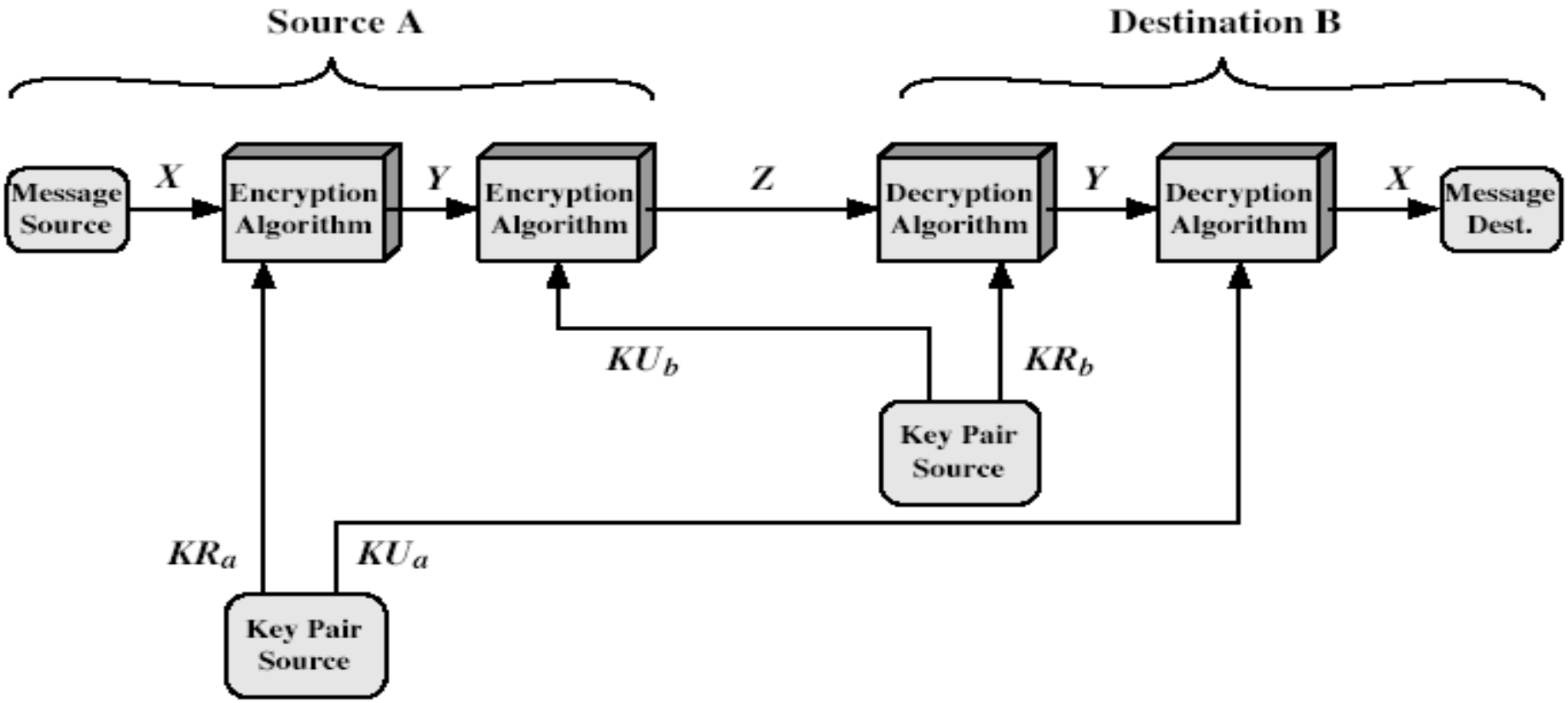


Figure 9.4 Public-Key Cryptosystem: Secrecy and Authentication

محاسبه نمای گسسته

- برای محاسبه $a^b \pmod N$ الگوریتمهای متفاوتی ابداع شده است...

  – فرض کنید $b_k b_{k-1} \ldots b_0$ نمایش مبنای ۲ عدد b باشد. توجه کنید که $k=\log(b)$

  – بنابراین تعداد k عمل محاسباتی بشرح خواهیم داشت:

$$a^b = a^{\sum_{b_i \neq 0} b_i . 2^i} = \prod_{b_i \neq 0} a^{b_i . 2^i} = \prod_{b_i \neq 0} \left(a^{2^i} \right)^{b_i}$$

$$a^b \bmod n = \left[\prod_{b_i \neq 0} a^{2^i} \right] \bmod n = \left[\prod_{b_i \neq 0} \left(a^{2^i} \bmod n \right) \right] \bmod n$$

الگوریتم توان و ضرب

- بر این مبنا میتوان الگوریتم زیر را طراحی نمود:

$c \leftarrow 0; d \leftarrow 1$
for $i \leftarrow k$ **downto** 0
do $c \leftarrow c \times 2$
$d \leftarrow d^2 \bmod n$
if $b_i = 1$
then $c \leftarrow c + 1$
$d \leftarrow (d \times a) \bmod n$
return d

RSA

- ✓توسط Rivest-Shamir -Adleman در سال 1977 در MIT ارائه شد
- ✓ اين روش كه چگونگي آن در زير تشريح شده است بنام روش RSA (مخفف اسامي آنها) مشهور است و بطرز فزاينده اي از آن استفاده مي‌شود
- ✓مشهورترين و پركاربردترين الگوريتم رمزگذاري كليد عمومي
- ✓مبتني بر توان رساني پيمانه ايي
- ✓استفاده از اعداد طبيعي خيلي بزرگ
- ✓امنيت آن ناشي از دشوار بودن تجزيه اعداد بزرگ، كه حاصلضرب دو عامل اول بزرگ هستند، مي باشد.
- ✓ متني كه بايد رمز شود به بلوكهايي تقسيم مي‌شود
- ✓مستندات مربوط به آن تحت عنوان PKCS استاندارد شده است.

نمادگذاری RSA

- N : پیمانه محاسبات
- e: نمای رمزگذاری
- d: نمای رمزگشایی
- M: پیام ، عدد صحیح متعلق به Z_N^*
- تابع RSA: $x \to x^e \bmod N$
- تابع معکوس: $x \to x^d \bmod N$

Key Generation

Select p, q	p and q both prime
Calculate $n = p \times q$	
Calculate $\phi(n) = (p - 1)(q - 1)$	
Select integer e	$\gcd(\phi(n), e) = 1; \quad 1 < e < \phi(n)$
Calculate d	$d = e^{-1} \bmod \phi(n)$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

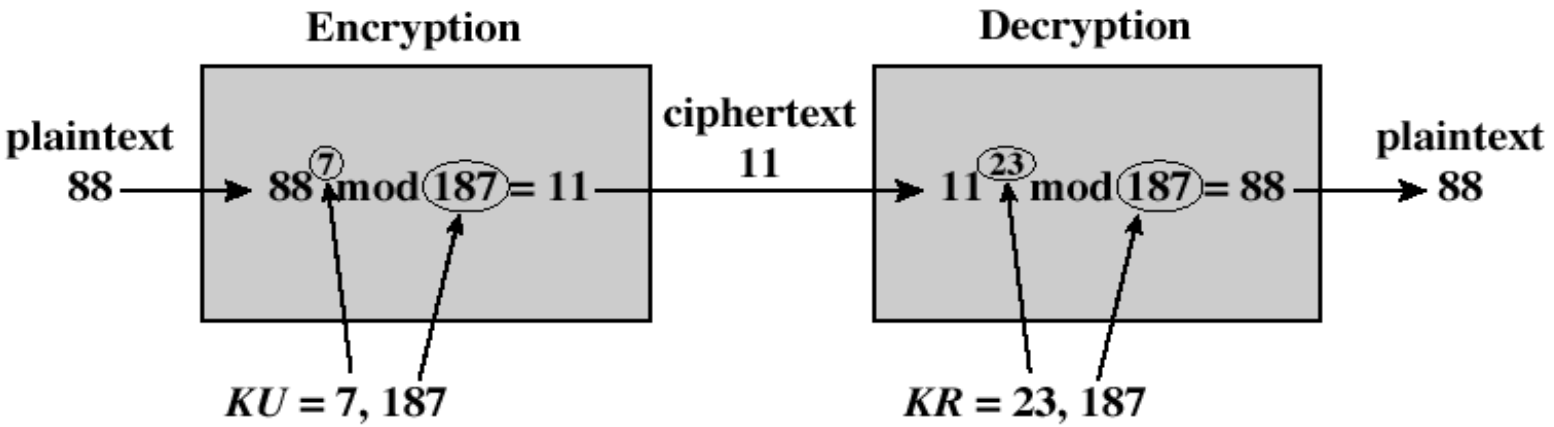
Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod{n}$

Decryption

Ciphertext:	C
Plaintext:	$M = C^d \pmod{n}$

- هم فرستنده و هم گيرنده مقدار N را مي‌دانند
- فرستنده مقدار e را مي‌داند
  - ✓ كليد عمومي : (N , e)
- تنها گيرنده مقدار d را مي‌داند
  - ✓ كليد خصوصي : (N, d)
- نيازمنديها:
  - ✓ محاسبه M^e و C^d آسان باشد
  - ✓ محاسبه d با دانستن كليد عمومي غيرممكن باشد



p = 17, q = 11, n = p*q= 187

Φ(n) = 16*10 =160, pick e=7, d.e=1 mod Φ(n) ➔ d = 23

نحوه تعریف کلیدهای عمومی و خصوصی

1- دو عدد بزرگ (هر چه بزرگتر بهتر) اول به نام هاي p و q را انتخاب مي كنيم، بهتر است اين اعداد از لحاظ سايز نزديک به يكديگر باشند.

2- عدد ديگري بنام n را معادل با حاصلضرب p در q تعريف مي كنيم : $n = p \times q$

3- عدد چهارم يعني m را معادل حاصلضرب p-1 در q-1 تعريف مي كنيم : $m = (p-1) \times (q-1)$

4-عدد e را كه از m كوچكتر است آنگونه پيدا مي كنيم كه بزرگترين مقسوم عليه مشترک اين دو يک باشد به عبارتي نسبت به هم اول باشند.

5- عددي مانند d را پيدا كنيم كه باقيمانده حاصلضرب d در e تقسيم بر m مساوي عدد 1 باشد، يعني :
$(d \times e) \bmod m = 1$

حال پس از طي اين مراحل مي توانيم از e و n بعنوان كليد عمومي و از d و n بعنوان كليد اختصاصي استفاده كنيم.

مثالی دیگر از نحوه تعریف کلید های عمومی و خصوصی

11=p
3=q

N=33

M=(p-1)*(q-1)=20

E=3

(d*e)mod m=1 7*3=21

کلید عمومی(3,33) کلید خصوصی(33,7)

اعداد 1 تا 32 و محاسبه رمزگذاری عمومی انها با این کلید عمومی خصوصی

m	۰ ۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ ۱۰ ۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶
c	۰ ۱ ۸ ۲۷ ۳۱ ۲۶ ۱۸ ۱۳ ۱۷ ۳ ۱۰ ۱۱ ۱۲ ۱۹ ۵ ۹ ۴
m	۱۷ ۱۸ ۱۹ ۲۰ ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰ ۳۱ ۳۲
c	۲۹ ۲۴ ۲۸ ۱۴ ۲۱ ۲۲ ۲۳ ۳۰ ۱۶ ۲۰ ۱۵ ۷ ۲ ۶ ۲۵ ۳۲

رمزگذاري كليد عمومي (Public Key Cryptography)

- قبل از آنكه روش رمزگشائي را تشريح كنيم الگوي رمزنگاري RSA را بصورت جمع‌بندي شده ارائه مي‌دهيم:

- **الف) رشته اي كه بايد رمز شود ، به بلوكهاي K كاراكتري تبديل مي‌شود.**

- **ب) هر بلوك طبق قاعدة دلخواه به يك عدد صحيح تبديل مي‌شود. (P_i)**

- **ج) با جفت عدد صحيح (e,n) براي تمام بلوكها اعداد جديدي طبق رابطه زير بدست مي‌آيد:**

$$C_i=(P_i)^e \bmod n$$

رمزگذاري كليد عمومي (Public Key Cryptography)

- **د) كدهاي Ci ، بجاي كد اصلي ارسال مي‌شود**

- نكته اساسي در اين الگو آنست كه براي رمزگشائي كدها بايد عددي مثل d پيدا شود كه در رابطه زير صدق كند:

$$(x^{e.d}) \bmod n = x$$

- با چنين عددي خواهيم داشت:

$$Pi = (Ci^{d}) \bmod n$$

- يعني مشابه عمل رمزنگاري مجدداً كدهاي رمز به توان d رسيده ، باقيماندة آن بر n محاسبه خواهد شد. كدهاي حاصل دقيقاً همان كدهاي اوليه هستند

رمزگذاري **كليد عمومي** (Public Key Cryptography)

- به كليد (e,n) كه با آن متن رمز مي‌شود "**كليد عمومي**" (Public key) و به كليد (d,n) كه با آن متن از رمز خارج مي‌شود "**كليد خصوصي**" (Private key) اطلاق مي‌شود

- قبل از آنكه مثالي ديگر ارائه بدهيم اجازه بدهيد روش انتخاب و معيارهاي d , e را كه توسط ابداع كنندگانِ اين روش پيشنهاد شده است ، معرفي كنيم:

- **الف) دو عدد اوّل دلخواه (ولي بزرگ) p , q انتخاب كنيد. (براي كاربردهاي عملي اگر اين اعداد صد رقمي باشند اطمينان بخش خواهد بود - يعني از مرتبه 10^{100} باشد-)**

رمزگذاري كليد عمومي (Public Key Cryptography)

- نكاتي كه در رمزنگاري بايد رعايت شود آنست كه كدهاي P_i كه به هر بلوك نسبت مي‌دهيم بايد باشد $0< P_i <n$
- بنابراين اگر بلوكها را بصورت رشته هاي k بيتي مدل مي‌كنيد بايد شرط $2^k \leq n$ برقرار باشد
- براي يك مثال آموزشي فرض كنيد بخواهيم رشته "SUZANNE" را رمز نمائيم
- براي راحتي كار مجبوريم كليدها را بسيار كوچك بگيريم ولي دقت داشته باشيد در عمل اينطور نيست:

رمزگذاري كليد عمومي (Public Key Cryptography)

- **الف) دو عدد اول p=3 و q=11 را انتخاب مي‌كنيم**
- **ب) عدد n=33 و m=20 بدست مي‌آيند**
- **ج) عدد 7 كه نسبت به m اول است را براي d انتخاب مي‌نمائيم**
- **د) بايد عدد e بگونه اي پيدا شود كه رابطة 7*e mod 20=1 برقرار باشد اين عدد را 3 انتخاب كرده ايم. (عدد 23 هم قابل قبول است)**

رمزگذاري كليد عمومي (Public Key Cryptography)

- پس داريم :

كليد عمومي(e,n)=(3,33)
كليد خصوصي(d,n)=(7,33)

- براي آشنايي با مراحل كار به شكل بعد دقت نمائيد

- بدليل آنكه n عدد كوچكي است و بايد $P_i<33$ باشد، مجبوريم بلوكها را يك كاراكتري فرض كرده و به A عدد 1 ، به B عدد 2 نسبت داده و بهمين ترتيب كاراكترها را به عدد صحيح تبديل نمائيم

رمزگذاري کليد عمومي (Public Key Cryptography)

سمبولهای متن	عدد P_i	محاسبهٔ P^3	$P^3 \bmod 33$	محاسبهٔ C^7	$C^7 \bmod 33$
S	19	6859	28	13492928512	**19**
U	21	9261	21	1801088541	**21**
Z	26	17576	20	1280000000	**26**
A	01	1	1	1	**1**
N	14	2744	5	78125	**14**
N	14	2744	5	78125	**14**
E	05	125	26	8031810176	**5**

رمزنگاری رمزگشایی

رمزگذاري كليد عمومي (Public Key Cryptography)

- همانگونه كه اشاره شد در عمل p و q صد رقمي انتخاب مي‌شوند. يعني

$$q \approx 10^{00} \quad , \quad p \approx 10^{00}$$

- بنابراين مقدار n از مرتبه 10^{200} (دويست رقمي) خواهد بود

- سؤال آنست كه عدد صحيح مربوط به بلوك هاي P_i كه بايد از n كوچكتر باشند چند بيتي خواهند بود؟

$$n < 10^{200} \approx (10^{200} \approx 2^{664}) \Rightarrow n < 2^{664}$$

- پس هر بلوك متن بايستي حداكثر 664 بيت يا معادل 83 كاراكترِ هشت بيتي باشد

رمزگذاري كليد عمومي (Public Key Cryptography)

- ممكن است تاكنون ذهن شما مشغول اين نكته شده باشد كه چگونه مي‌توان اعداد با اين عظمت را به توان رساند

- نكته ظريفي كه وجود دارد آنست كه براي محاسبة $P^e \bmod n$ لازم نيست كه اول P به تعداد e بار در خودش ضرب شود و بعد باقيمانده آن بر n بدست آيد

- براي روشن شدن قضيه به الگوي زير دقت كنيد:

رمزگذاري كليد عمومي (Public Key Cryptography)

$$7^3 \bmod 5=((7 \bmod 5)* 7^2) \bmod 5=(2* 7^2) \bmod 5=((2*7 \bmod 5)* 7) \bmod 5=((4*7) \bmod 5=3$$

• فرض كنيد بخواهيم A را به توان E برسانيم و بسط E در مبناي دودوئي بصورت زير باشد:

$$E=(e_{k-1},\ldots.e_0)_2=\sum_{i=0}^{k-1} e_i 2^i$$

$$A^E=A^{\sum_{i=0}^{k-1} e_i 2^i}=A^{2^{k-1}.e_{k-1}}\times\ldots..\times A^{2^1.e_1}\times A^{2^0.e_0}$$

•كه اين محاسبات داراى پيچيدگى k=logE ميباشد،

رمزگذاري كليد عمومي (Public Key Cryptography)

- اگر دقت داشته باشيد الگوريتم فوق با مثال قبلي ($7^3 \bmod 5$) معادل خواهد بود
- بنابراين مشكل حادّي در عمليات محاسبه كدهاي رمز RSA و همچنين رمزگشائي آن وجود ندارد
- به ياد داشته باشيد كه كليد رمزگذاري (e,n) يك كليد عمومي است و دلايلي بر سرّي و محرمانه ماندن آن وجود ندارد در حالي كه كليد رمزگشائي (d,n) كليد اختصاصي است و بايد سرّي باشد
- براي شكستن رمز RSA بايد مقدار d را از (e,n) به دست آورد

رمزگذاري كليد عمومي (Public Key Cryptography)

- براي بدست آوردن d ابتدا بايد n را به عوامل اول تجزيه كرد تا بتوان p، q و m و نهايتاً d را بدست آورد

- با توجه به آنكه n معمولا دويست رقمي است با كامپيوترهاي معمولي براي تجزيه چنين عددي چهار ميليون سال طول خواهد كشيد!

- به جدول بعد نگاه كنيد فرض كنيد كامپيوتري هر عمل را در يك ميكروثانيه انجام بدهد اين جدول زمان تجزيه يك عدد را به عوامل اوّل بر حسب تعداد ارقامِ عدد مشخص كرده است

- گرچه تحقيق بر روي تجزية اعداد به عوامل اول ادامه دارد ولي هيچ الگوريتم كارآمدتري كه بتواند زمانهاي جدول فوق را كاهش بدهد پيدا نشده است و بهمين دليل بطور فراگير از آن استفاده ميشود

روش تجزیه n=pq فعلا معلوم نیست، شناخت اعداد دوم!!

حال آیا روش فهم اول بودن یک عدد مشخص است؟

```
For i=2 to sqrt(n)
        If prime(i)==1 // prime(i) is O(log(i)^4) توسط دانشجویان هندی
            if n mod i=0 print I, n/I
End for
```

Omega(sqrt(n))

الگوریتم Pohling-Hellman

- مفروضات:
  - قضیه اولر gcd(a,n)=1 آنگاه $a^{\phi(n)} = 1 \mod d$
  - N عدد اول میباشد.
$$d \times e \equiv 1 \mod \phi(N)$$
$$d \times e = k * \phi(N) + 1$$
- رمزگذاری:

$$C = M^e \mod N$$

- رمزگشایی

$$C^d \mod N = \left(M^e\right)^d \mod N$$
$$= M^{ed} \mod N = M^{k*\phi(n)+1} \mod N$$
$$= (M^{\phi(n)})^k * M \mod N = M \mod N = M$$

مبانی ریاضی RSA

- p و q دو عدد اول میباشند.
- $\varphi(N)$: تعداد اعداد(کوچکتر ازN) که نسبت به N اول است.

$$N = p \times q$$
$$\varphi(N) = (p-1) \times (q-1)$$
$$\gcd(\varphi(N), e) = 1$$
$$d \times e \equiv 1 \bmod \varphi(N)$$
$$C = M^e \bmod N$$
$$M = C^d \bmod N = (M^e)^d \bmod N$$

زمان بدست آوردن تجزیه n به p*q

تعداد ارقام	زمان محاسبه
۵۰	۴ ساعت
۷۵	۱۰۴ روز
۱۰۰	۷۴ سال
۲۰۰	چهار میلیون سال
۳۰۰	$10^{15} \times 5$ سال
۵۰۰	$10^{25} \times 4$ سال

شکستن الگوریتم

آیا متنی که توسط الگوریتم RSA بصورت رمز شده و مخفی درآمده است قابل شکسته شدن است؟ این سئوالی است که اغلب راجع به همه روشهای رمز کردن اطلاعات پرسیده می شود. واقعیت آن است که همه روشهای رمز کردن قابل شکستن است، اما نکته مهم آن است که در چه مدت زمان و با چه امکاناتی این اطلاعات باید رمزگشایی شوند. در ارتباط با الگوریتم RSA باید گفت روشهای محدودی برای شکستن متن رمز شده توسط آن وجود دارد که در اینجا به مواردی از آن اشاره می کنیم.

تجزیه n به عوامل اول

اولین روش آن است که بتوان کلید خصوصی را حدس زد و یا پیدا کرد، در این صورت هکر می تواند تمامی متن های تهیه شده با کلید عمومی را رمزگشایی کند و بخواند و یا می تواند از امضای الکترونیک صاحب کلید استفاده کند. فرض را بر این می گذاریم که فردی که قصد حدس زدن کلید خصوصی را دارد، از جمله افرادی است که کلید عمومی را دارا است. در این حالت او n و e را در دسترس دارد.

حال اولین قدم برای این آقای هکر آن است که بتواند از روی عدد n عاملهای p و q را حدس بزند. این مشکلترین قسمت کار است که محاسبات ریاضی و بررسی های انجام داده شده نشان می دهد اگر عدد n مثلا" ۱۵۵ رقم داشته باشد (RSA-155) در آن صورت با قوی ترین کامپیوتر های موجود بیش از ۷ ماه زمان لازم است تا بتوان عوامل اول تشکیل دهنده n را مشخص کرد. الگوریتم های ریاضی بدست آمده نشان می دهد که اعداد بزرگ اگر عوامل اول کوچکتری داشته باشند، ساده تر تجزیه می شوند تا اعداد بزرگی که عوامل اول بزرگتری دارند.

بدست آوردن روش موثر برای محاسبه ریشه eام

با توجه به روش رمز کردن شما با داشتن کلید عمومی n و e و استفاده از فرمول $C = M^e \bmod n$ می توانید حروف را رمز کنید. اما با نگاهی به فرمول می توان دریافت که کافی است شما بتوانید ریشه eام C mod n را بدست آورید در آن صورت شما می توانید به عدد m نزدیک شوید و کاراکتر اولیه برسید.

نکته مهم آن است که شما در اینجا کلیدای را کشف نکرده اید و فقط توانسته اید کاراکتر را بدست آورید، ضمن آنکه بنظر نمی رسد که در حال حاضر کسی از این روش برای رمز گشایی استفاده کند چرا که به مراتب دشوار تر از روش اول است. این روش فقط برای مواردی که e عدد کوچک باشد کاربرد آزمایشگاهی و آموزشی دارد و در رمزکردن های معمولی به هیچ وجه مورد استفاده موفقیت آمیز حتی در زمانهای طولانی ندارد.

حدس زدن پیام

برای باز کردن رمز پیامهایی که با الگوریتم RSA رمز شده اند، روشهای محاسبه ریاضی عملا" راه به جایی نمی برند، این است که در مواردی که متن کوچک باشد شاید حدس زدن متن اصلی ساده ترین روش برای رمزگشایی باشد. ارسال پیام های کوتاه دو یا سه کلمه ای و تشخیص ساده آنها توسط هکر می تواند به او کمک کند که از روی پیام رمزگشایی شده کلید خصوصی شما را حدس بزند. در این گونه موارد کافی است تعداد زیادی کلمات یا بیت های اتفاقی (Random) در انتهای پیام بگذارید تا هکر نتواند پیام شما را حدس بزند

راههای مقابله با حمله زمانی به RSA

- استفاده از توان رساندن با زمان ثابت محاسباتی.
  - تابع باید به ازای همه ورودیها زمان ثابتی به طول بیانجامد

- قرار دادن اعمال اضافی و گمراه کننده در بین محاسبات
  - ضرب کردن متن رمزشده در یک عدد تصادفی قبل از عملیات به توان رسانی
  - اضافه کردن تاخیرهای تصادفی

چرا NSAده میلیون دلار به توسعه‌دهندگان RSAپرداخت کرد؟

۲ دی, ۱۳۹۲ - ۱۴:۲۲

فعالیت‌های آژانس امنیت ملی آمریکا موسوم به NSAدر چند ماه اخیر به شدت تحت تأثیر افشاگری‌های کارمند سابق این آژانس، ادوارد اسنودن می‌باشد.

ادوارد اسنودن به تازگی اعلام کرده است که NSAمبلغ ۱۰ میلیون دلار به توسعه‌دهندگان الگوریتم رمزنگاریِ RSAپرداخت کرده است. وی اذعان داشت که این مبلغ به در قبال قرارِدادی بوده است که توسعه‌دهدگان این الگوریتم را موظف می‌کرد، نقاط ضعف عمدی و غیرقابل شناسایی را در این الگوریتم قرار دهند که راه را برای جاسوسی‌های NSAهموارتر می‌کند.

چرا NSAده میلیون دلار به توسعه‌دهندگان RSA پرداخت کرد؟

توسعه‌دهندگان این الگوریتم مشهور رمزنگاری شب گذشته وجود چنین قراردادی را به کلی تکذیب کردند و توضیح داده‌اند که به عنوان اعضای جامعه امنیتی و حتی به عنوان شرکت‌های خارج از مجموعه‌ی آژانس به منظور فعالیت‌های امنیتی با این آژانس همکاری داشته‌اند، اما هیچ گاه چنین قراردادی بین آن‌ها وجود نداشته و تمامی قراردادهای این شرکت با NSA به صورت عمومی اعلام شده است.

چرا NSAده میلیون دلار به توسعه‌دهندگان RSAپرداخت کرد؟

علی‌رغم گفته‌های این شرکت، چندی پیش نیز اخباری منتشر شد مبنی بر این که NSAاز سال‌ها پیش برنامه‌های گسترده‌ای برای تضعیف استانداردهای عمومی امنیت از طریق سازمان ملی استاندارهای آمریکا، NISTداشته است.

آیا نمی‌توان فرض کرد که مطابق همچنین قراردادی نرم‌افزار رمزنگاری Bsafeکه مبتنی بر همین الگوریتم است عمداً دچار ضعف بود؟ و یا حتی زمانی که کارشناسان امنیتی نسبت به Dual ECوضعف‌های آن انتقاد می‌کردند و این توسعه‌دهندگان ساکت بودند و در حال کسب درآمد از طریق قرارداد با NSAبوده‌اند؟ (Dual ECابزار اصلی تولید اعداد تصادفی است که توسط NISTتأیید شده بود و به تازگی مشخص شد که ضعف آن از وجود درپشتی حاصل می‌شود که برای NSAباز گذشته شده است!)

چرا NSAده میلیون دلار به توسعه‌دهندگان RSAپرداخت کرد؟

توسعه‌دهندگان RSAدر توضیح این مسائل اذعان داشته‌اند که استفاده از Dual ECدر نرم‌افزار Bsafeدر سال ۲۰۰۴ صورت گرفته است و این در شرایطی بود که همه‌ی افراد فعال حوزه‌ی امنیت خواستار تعویض Dual ECو استفاده از نسخه‌ای جدیدتر و قوی‌تر بودند و این در شرایطی بود که هنوز صنایع اعتماد خود را به NSAاز دست نداده بودند و Dual ECمطابق با استانداردهای آن دوره امن بوده است.

در سال ۲۰۰۶ مقاله‌ای توسط Berry Schoenmakersو Andrey Sidorenkoاز دانشگاه Eindhoven University of Technologyبا عنوان

Cryptanalysis of the Dual Elliptic Curve Pseudorandom Generator

چاپ شده است که ثابت می‌کند الگوریتم DEC PRGبرای تولید اعداد تصادفی به شدت ناامن است.

اما علی‌رغم این اثبات در ۷ سال پیش، این الگوریتم تا یک ماه گذشته توسط NISTمعتبر شمرده می‌شد و سپس به دلیل فاش شدن در پشتی آن توسط این مؤسسه استاندار کنار گذاشته شد.

واکنش شدید F-Secure در مقابل همکاری RSA با NSA

- ۳ دی, ۱۳۹۲ - ۱۶:۰۹
- همان‌طور که در اخبار اشاره شد، به تازگی افشا شده است که توسعه‌دهندگان الگوریتم رمزنگاریِ RSA در قبال دریافت ۱۰ میلیون دلار اقدام به استفاده از مولد اعداد تصادفی‌ای در الگوریتم خود داشته‌اند که از سوی NSA تعبیه شده و دارای یک درپشتی برای جاسوسی‌های NSA بوده است.
- علی‌رغم این که کارشناسان امنیتی سال‌ها به این مولد اعداد تصادفی انتقاداتی وارد می‌کردند، توسعه‌دهندگان RSA حاضر به تعویض آن نمی‌شدند تا این که پس از علنی شدن وجود در پشتی در این مولد، بالاخره پس از سال‌ها RSA این الگوریتم را کنار گذاشت.

اما این‌که RSA در مقابل این کار مبلغ ۱۰ میلیون دلار نیز به عنوان حق‌الزحمه از NSA دریافت کرده است خشم فعالان حوزه‌ی امنیت فناوری اطلاعات را برانگیخته و آن‌ها را وادار به واکنش کرده است.

واکنش شدید F-Secure در مقابل همکاری RSA با NSA

Mikko Hypponenمدیر تحقیقات شرکت امنیتی F-Secureکه به علت رفتار صریح و البته بیش‌تر قدرت بالا در تحلیل مسائل امنیتی مشهور است در نامه‌ای سرگشاده خطاب به مسئولان شرکت امنیتی RSAکه وظیفه‌ی برگزاری هرساله‌ی کنفرانس رمزنگاری RSAرا نیز برعهده دارند، اینطور نوشته است:

«من از سال ۱۹۹۱ در حوزه‌ی امنیت مشغول به کار می‌باشم. در سال‌های اخیر البته سخرانی‌های عمومی نیز ارائه می‌دهم، درواقع من ۸ بار در کنفرانس‌های RSAدر آمریکا، ژاپن و اروپا سخنرانی کرده‌ام و عکس من را به عنوان «خبره‌ی صنایع» در دیوار سالن‌های کنفرانس استفاده کرده‌اید.

در تاریخ ۲۰ دسامبر، خبرگزاری رویترز، داستانی را در مورد اینکه شما در قبال دریافت ۱۰ میلیون دلار اقدام به استفاده از مولد اعداد تصادفیِ NSAکرده‌اید و این نکته که شما اقدام به نصب این مولد به صورت گزینه‌ی پیش‌فرض در یکی از محصولات خود داشته‌اید را منتشر کرد. شرکت شما در مقابل این خبر واکنش نشان داد اما در واقع شما این ادعا را از پایه بی‌اساس نخوانده‌اید و دریافت پول را تکذیب نکرده‌اید. در نهایت چندی پیش پس از آن که مشخص شد این مولد اعداد تصادفی دارای یک در پشتی تعبیه شده برای NSA است شما آن را کنار گذاشتید، علی‌رغم این که کارشناسان امنیتی در طول سال‌ها بارها تذکر داده بودند که این مولد دارای اشکالات اساسی است اما شما توجهی به این انتقادها نمی‌کردید و از مولدی استفاده می‌کردید که در واقع یک راه جاسوسی برای NSAباز می‌گذاشت.

در مقابل این عمل شما، سخرانی خود در کنفرانس RSAدر فوریه‌ی سال ۲۰۱۴ در سان‌فرانسیسکو را لغو اعلام می‌کنم.

سخرانی من با موضوع «دولت‌ها به عنوان نویسندگان بدافزار» در این کنفرانس ارائه نخواهد شد. من انتظاری ندارم که شرکتِ میلیاردر شما و یا کنفرانس میلیونیِ شما در نتیجه‌ی رابطه‌ای که با NSAدارد، ضرری را متحل شده باشد و البته انتظاری هم از سایر سخنرانان ندارم که مقالات و سخنرانی‌های خود را لغو کنند. غالب سخنرانان کنفرانس شما، آمریکایی‌هایی هستند که مطمئناً مورد هدف جاسوسی‌های NSAنیستند و سایر افراد خارجی هدف این عملیات جاسوسی می‌باشند. به هر حال من هم یک خارجی هستم و از کنفرانس شما حمایت نخواهم کرد.»

لغو اعتراضی سخنرانی‌های کنفرانس RSA ادامه دارد

- پس از انتشار ارتباط RSA با آژانس امنیت ملی آمریکا موسوم به NSA، افراد زیادی در اعتراض به این رابطه سخنرانی‌های خود را در کنفرانس امنیتی RSA که یکی از بزرگ‌ترین کنفرانس‌های امنیتی سالانه آمریکاست لغو کرده‌اند.
- سخنرانی در این کنفرانس از افتخارات سخنرانان محسوب می‌شود و بسیاری از افراد مشهور حوزه‌ی امنیت هر ساله در این کنفرانس سخنرانی می‌کنند. کنفرانس سال ۲۰۱۴، طبق برنامه، اواخر فوریه در لس‌آنجلس برگزار خواهد.
- اما همان‌طور که از اخبار مطلع شده‌اید، ماه گذشته در افشاگری‌هایی مشخص شد که شرکت امنیتی RSA، سال‌ها قبل طی قراردادی که با NSA داشته است در قبال دریافت ۱۰ میلیون دلار موظف شده است الگوریتم رمزنگاری خود را به نحوی تغییر دهد که یک درپشتی برای NSA در آن تعبیه شده باشد.
- شرکت RSA در مقابل این خبر واکنش نشان داد اما در واقع این شرکت این ادعا را از پایه بی‌اساس نخوانده و دریافت پول را تکذیب نکرده است.
- در نهایت چندی پیش پس از آن که مشخص شد این مولد اعداد تصادفی دارای یک در پشتی تعبیه شده برای NSA است بالاخره RSA آن را کنار گذاشت، علی‌رغم این که کارشناسان امنیتی در طول سال‌ها بارها تذکر داده بودند که این مولد دارای اشکالات اساسی است اما RSA توجهی به این انتقادها نمی‌کرد و از مولدی استفاده می‌کرد که در واقع یک راه جاسوسی برای NSA باز می‌گذاشت. اما این تکذیبه تأثیر چندانی در خبرگزاری‌ها نداشت و خشم فعالان امنیتی را کم نکرد.
- **Josh Thomas یکی از روسای شرکت امنیتی ATREDIS نیز سخنرانی خود را لغو کرده و در توییتر خود اعلام کرده است که موضوع سخنرانی‌اش را از طریق وبلاگش منتشر می‌کند.**
- **Chris Soghoian از شرکت امنیتی ACLU نیز در توییتر خود اعلام کرد که سخنرانی‌اش در این کنفرانس لغو شده است.**
- **Adam Langley و Chris Palmer که در گوگل مشغول به کار هستند، نیز اعلام کردند هر دو سخنرانی آن‌ها لغو شده است.**
- **Marcia Hoffman از EFF نیز اعلام کرد که سخنرانی‌اش لغو است و در نهایت Alex Fowler از شرکت موزیلا نیز به همین ترتیب سخنرانی‌اش را لغو کرده است.**

لغو اعتراضی سخرانی‌های کنفرانس RSAادامه دارد

- برخی دیگر از سخن‌رانان اما شواهد ارائه‌شده را کافی نمی‌دانند و لغو سخن‌رانی‌ها را بهترین انتخاب در این شرایط نمی‌بینند.
- **کیفیت کنفرانس سال جاری، هنوز مشخص نیست، اما با احتمال بالایی افراد زیادی در صف انتظار برای ارائه‌ی سخن‌رانی در این کنفرانس هستند و لغو سخن‌رانی‌ها این کنفرانس را با مشکل جدی مواجه نمی‌کند، اما کیفیت این سخن‌رانی‌ها تا روز ارائه مشخص نخواهد شد.**