Related Randomness Attacks for Public Key Encryption

Kenneth G. Paterson, Jacob C. N. Schuldt, Dale L. Sibborn

Information Security Group, Royal Holloway, University of London

March 2014

Randomness Failures

- Most modern cryptographic primitives are heavy consumers of randomness.
- These schemes are provably secure when uniform randomness is available.
- Random Number Generators (RNGs) often fail to provide high-quality randomness in practice due to poor design, insufficient entropy, bugs, etc.
- Randomness failures can be catastrophic.

Hackers Describe PS3 Security As Epic Fail, Gain Unrestricted Access

by Mike Bendel on December 29, 2010 @ 11:19 am

Home > Security

News

Microsoft confirms that XP contains random number generator bug

A fix is coming next year in SP3; Vista, Server 2003 and Server 2008 immune

By Gregg Keizer

November 21, 2007 12:00 PM ET

Computerworld - Windows XP, Microsoft's operating system, sports the same encryption ... recently disclosed in Windows 2000 ... Wednesday.

SECURITY

Android bug batters Bitcoin wallets

Old flaw, new problem

By Richard Chirgwin, 12th August 2013

Redefining enterprise software

9

RELATED STORIES

Users of Android Bitcoin apps have woken to t... pseudo random number generation bug has b... us...

The New York Times

Business Day

Technology

WORLD U.S. N.Y. / REGION BUSINESS TECHNOLOGY SCIENCE HEALTH

Flaw Found in an Online Encryption Method

By JOHN MARKOFF

Published: February 14, 2012

127 Comments

... A team of European and American ... have discovered an unexpected ... widely used worldwide for online ... Internet services intended to

Debian OpenSSL Security Flaw

May 26th, 2008

Hint to 99.98% of you reading this: Skip to the next story.

Reader GP submitted **Alarming Open-Source Security Holes**, by Simson Garfinkel, which is a human readable version of **this Debian Security Advisory**:

Luciano Bello discovered that the random number generator in Debian's openssl package is predictable. This is caused by an incorrect Debian-specific change to the openssl package (CVE-2008-0166). As a result, cryptographic key material may be guessable.

What Should We Do?

- Ideally, we should design better RNGs.
- Unfortunately, such randomness failures seem to be endemic and are hard to eliminate.
- Hence, we must do the next best thing.
- It is desirable to study models that capture these failures and design schemes that are secure in these models.

Previous Work

- Bellare et al. introduced the *Chosen Distribution Attack*
  - adversaries specify a joint distribution on messages and randomness
- Yilek studied *Reset Attacks*
  - adversary can see challenge encryptions using repeated randomness
- Ristenpart and Yilek studied several randomness attacks
  - adversary can repeat, predict, or choose the randomness

Our Model

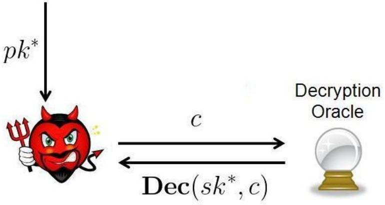
- We introduce the RRA game to model Related Randomness Attacks
- The Related Randomness Attack game is an indistinguishability game similar to IND-CPA/CCA, but additionally adversaries can
  1. force the reuse of random coins (as in the Reset Attack setting)
  2. force the use of *functions* of those random coins (similar to the RKA setting)
- This framework can model
  1. encryption with a faulty RNG
  2. imperfect VM resets (due to clock synchronisation)

RRA-CCA Game

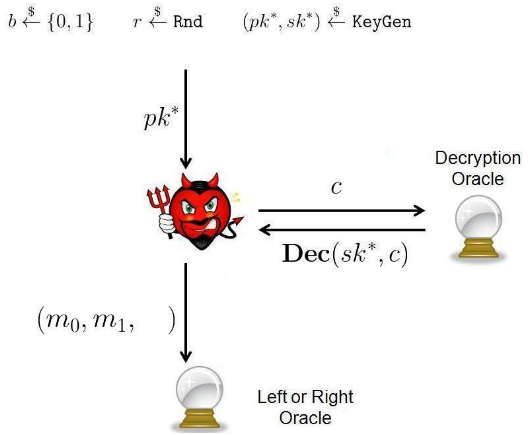
$$b \xleftarrow{\$} \{0,1\} \qquad r \xleftarrow{\$} \mathtt{Rnd} \qquad (pk^*, sk^*) \xleftarrow{\$} \mathtt{KeyGen}$$

RRA-CCA Game

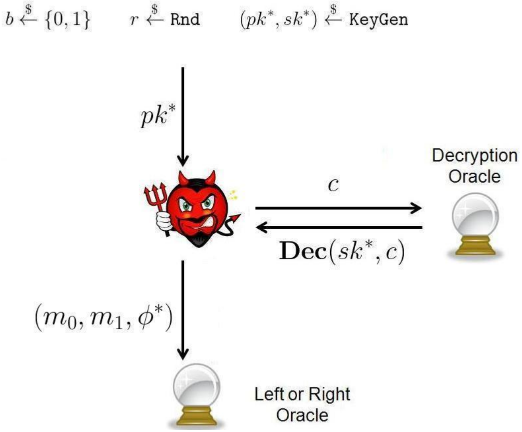
$b \xleftarrow{\$} \{0,1\}$ $\quad r \xleftarrow{\$} \texttt{Rnd}$ $\quad (pk^*, sk^*) \xleftarrow{\$} \texttt{KeyGen}$



RRA-CCA Game

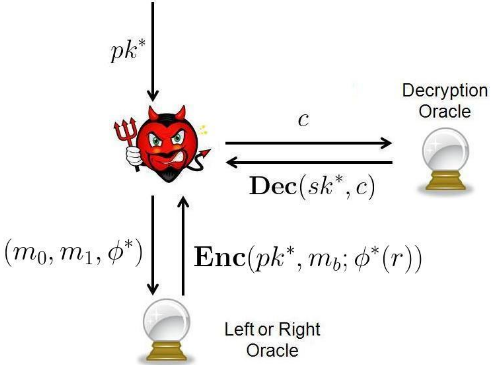

$b \xleftarrow{\$} \{0,1\}$
$r \xleftarrow{\$} \mathtt{Rnd}$
$(pk^*, sk^*) \xleftarrow{\$} \mathtt{KeyGen}$
pk^*
Decryption Oracle
c
$\mathbf{Dec}(sk^*, c)$
$(m_0, m_1, \quad)$
Left or Right Oracle

RRA-CCA Game


$b \xleftarrow{\$} \{0,1\}$
$r \xleftarrow{\$} \texttt{Rnd}$
$(pk^*, sk^*) \xleftarrow{\$} \texttt{KeyGen}$
pk^*
c
Decryption Oracle
$\mathbf{Dec}(sk^*, c)$
(m_0, m_1, ϕ^*)
Left or Right Oracle

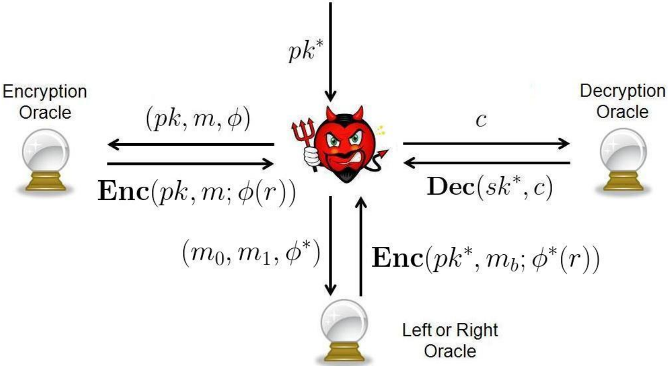
RRA-CCA Game

$b \xleftarrow{\$} \{0,1\} \qquad r \xleftarrow{\$} \texttt{Rnd} \qquad (pk^*, sk^*) \xleftarrow{\$} \texttt{KeyGen}$



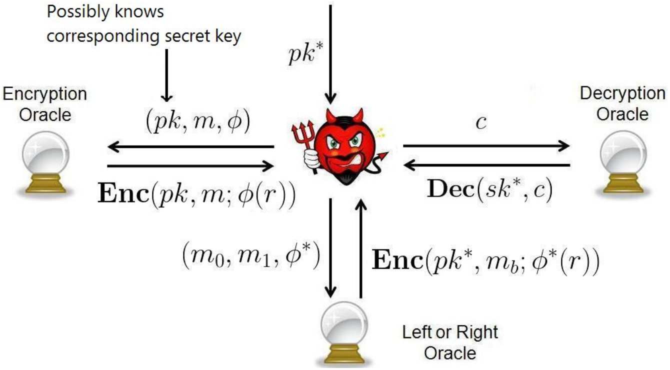
RRA-CCA Game

$b \xleftarrow{\$} \{0,1\} \qquad r \xleftarrow{\$} \mathtt{Rnd} \qquad (pk^*, sk^*) \xleftarrow{\$} \mathtt{KeyGen}$



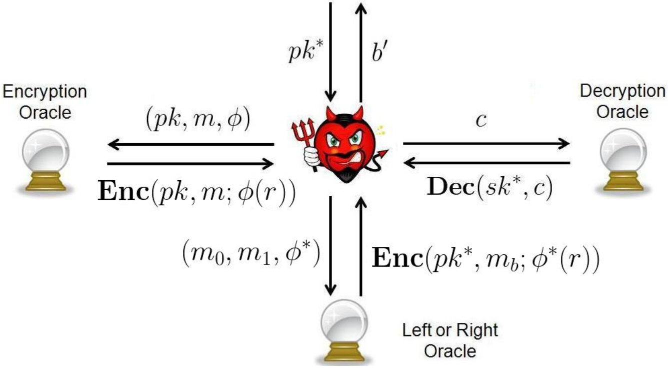
RRA-CCA Game

$$b \xleftarrow{\$} \{0,1\} \qquad r \xleftarrow{\$} \mathtt{Rnd} \qquad (pk^*, sk^*) \xleftarrow{\$} \mathtt{KeyGen}$$



RRA-CCA Game

$b \xleftarrow{\$} \{0,1\} \qquad r \xleftarrow{\$} \texttt{Rnd} \qquad (pk^*, sk^*) \xleftarrow{\$} \texttt{KeyGen}$



Trivial Wins

- Fixed randomness r is used in every encryption
  - encryption is essentially deterministic now.
- Hence, we inherit limitations similar to those of deterministic encryption.
- An example trivial win:

$$\frac{\textbf{LR}\text{ query }(m_0, m_1, \phi)\text{:}}{c \leftarrow \mathrm{ENC}(pk^*, m_b; \phi(r))} \qquad \frac{\textbf{LR}\text{ query }(m_0, \widetilde{m_1}, \phi)\text{:}}{c' \leftarrow \mathrm{ENC}(pk^*, m_b; \phi(r)).}$$

  If $c = c'$, the adversary outputs 0.
- An adversary that does not mount this (or a similar) kind of attack is called *equality pattern respecting*.

Φ-RRA-ATK Security

- We consider adversaries that are Φ-restricted (oracle queries only contain functions from the set Φ).
- We define the advantage of an adversary $\mathcal{A}$ against a scheme $\mathtt{PKE}$ as

$$\mathbf{Adv}^{\text{rra-atk}}_{\mathcal{A},\mathtt{PKE}}(\lambda) := 2 \cdot \mathbb{P}[\mathrm{RRA\text{-}ATK} \Rightarrow 1] - 1,$$

  where $\mathrm{ATK} \in \{\mathrm{CPA}, \mathrm{CCA}\}$, and the game outputs 1 if the adversary correctly guesses the bit b.
- A scheme $\mathtt{PKE}$ is Φ-$\mathrm{RRA\text{-}ATK}$ secure if the advantage of any polynomial time, equality pattern respecting, Φ-restricted adversary is negligible in the security parameter, λ.

Necessary Conditions on Φ

- For a scheme $\mathtt{PKE}$ to be $\Phi\text{-RRA-ATK}$ secure, we prove that the set Φ must satisfy the following two conditions:
  1. collision-resistance
  2. and output-unpredictability.
- The conditions are also necessary in the RKA setting.
- Security against function classes that do not satisfy these properties is impossible for **any** scheme.

Random Oracle Model

- In the Random Oracle model, we prove that collision-resistance and output-unpredictability are not only necessary, but they are **sufficient**.
- Consider the scheme $\texttt{Hash-PKE}$, built from a PKE scheme and a hash function H, that encrypts as follows:

$$\texttt{Hash-PKE.ENC}(pk, m; r) := \texttt{PKE.ENC}(pk, m; H(pk||m||r)).$$

- We prove that this scheme is Φ-RRA-ATK secure when the PKE scheme is IND-ATK secure and the set Φ is collision-resistant and output-unpredictable.

Standard Model Constructions

- We present two transforms for converting IND-secure schemes into RRA-secure schemes.
  1. The first transform requires a Related Key Attack secure PRF (RKA-PRF).
  2. The second utilises a Correlated Input Secure Hash function (CIS hash).

- The known instantiations of RKA-PRFs are secure against group-induced functions.
- Known CIS hash functions are selectively secure against polynomial functions.
- Our final construction is not a transform, but is a specific scheme that is secure against hard-to-invert functions.

Related Key Attack (RKA) Transform

- We show how to convert an IND-secure scheme $\mathtt{PKE}$ into an RRA-secure scheme $\widetilde{\mathtt{PKE}}$ via an RKA-PRF, F:

$$\boxed{\boxed{\text{IND-ATK PKE}}} + \boxed{\boxed{\text{RKA-PRF}}} \Rightarrow \boxed{\boxed{\text{RRA-ATK}}}.$$

- The key generation and decryption algorithms are unchanged. Encryption is as follows:

$$\widetilde{\mathtt{PKE.ENC}}(pk, m; r) := \mathtt{PKE.ENC}(pk, m; F_r(pk||m)).$$

- If F is a secure Φ-RKA-PRF, then the scheme described above is Φ-RRA-ATK secure.

RKA-PRF Limitations

- Unfortunately there are very few known RKA-PRFs.
- Bellare & Cash proved the existence of RKA-PRFs under the DDH and DLIN assumptions.
- The PRFs are only secure against group-induced functions ($\phi_a(r) = a * r$).
- The PRFs are not very practical and are only proofs-of-concept.
- Hence, we would like to find alternative solutions or stronger RKA-PRFs.

CIS Hash Transform

- We prove that

$$\boxed{\text{IND-ATK PKE}} + \boxed{\text{CIS Hash}} + \boxed{\text{PRF}} \Rightarrow \boxed{\text{RRA-ATK}}.$$

- $\widetilde{pk} = (pk, k)$, where k is a key for a CIS hash function h.
- The new encryption algorithm $\widetilde{\mathtt{PKE.ENC}}(\widetilde{pk}, m; r)$ is:

$$\begin{aligned} r' &\leftarrow h_k(r) \\ r'' &\leftarrow F_{r'}(\widetilde{pk}||m) \\ c &\leftarrow \mathtt{PKE.ENC}(pk, m; r''). \end{aligned}$$

- The scheme $\widetilde{\mathtt{PKE}}$ is Φ-RRA-ATK secure if the hash function is Φ-Correlated Input Secure and the adversary is restricted to using honestly-generated public keys (but the adversary is also given the secret keys).

CIS Hash Functions

- Goyal et. al gave a construction of a CIS hash that is selectively secure against uniform-output polynomial functions.
- The hash is defined as:

$$h_k(r) := g^{\frac{1}{k+r}},$$

where g generates a group $\mathbb{G}$ of prime order.

- This construction is secure based on the DDH assumption.

Security against hard-to-invert functions

- A collection of functions is δ-hard-to-invert if it is impossible to recover r with probability greater than δ when given $\phi_1(r), \ldots, \phi_q(r)$.
- The BHHO scheme is secure in the auxiliary input model (leakage of secret key).
- We consider a modified version of their scheme.
- In what follows, r_i denotes the ith bit of r.

Alg. $\mathtt{mBHHO.KEYGEN}(1^\lambda)$:

$g_1, \ldots, g_\lambda \leftarrow_\$ \mathbb{G}$
$x \leftarrow_\$ \mathbb{Z}_p$
$pk = \{g_i, g_i^x\}_{i=1,\ldots,\lambda}$
$sk = x$

Alg. $\mathtt{mBHHO.ENC}(pk, m)$:

$r \leftarrow_\$ \{0,1\}^\lambda$
$c_1 = \prod_{i=1}^{\lambda} g_i^{r_i}$
$c_2 \leftarrow m \cdot \prod_{i=1}^{\lambda} (g_i^x)^{r_i}$
$c = (c_1, c_2)$

RRA-CPA and Hard-to-Invert Functions

- In a slightly different model, we are able to prove that
  - if an adversary may only make one **LR** query, our modified BHHO scheme is Φ-RRA-CPA secure for any set of functions Φ that is sufficiently hard-to-invert.
  - we prove that a more complicated version of this scheme is secure when an adversary has *multiple* **LR** queries. Details are in the paper.
  - the proof uses similar techniques to leakage-resilient cryptography.

Conclusions

- We have introduced the RRA game that captures attacks not modelled by previous work.
- We have shown necessary conditions required of the class Φ to achieve RRA security. Furthermore, we proved that these conditions are **sufficient** in the Random Oracle model.
- We have developed connections with RKA-PRFs, CIS hash functions, and leakage resilience.
- Our RKA-PRF transform is provably secure against group-induced functions.
- Our CIS hash transform can protect against uniform-output polynomials.
- Our modified BHHO scheme is secure against hard-to-invert functions.

Thank you.

Questions?