On the Possibility of a Backdoor in the Micali-Schnorr Generator

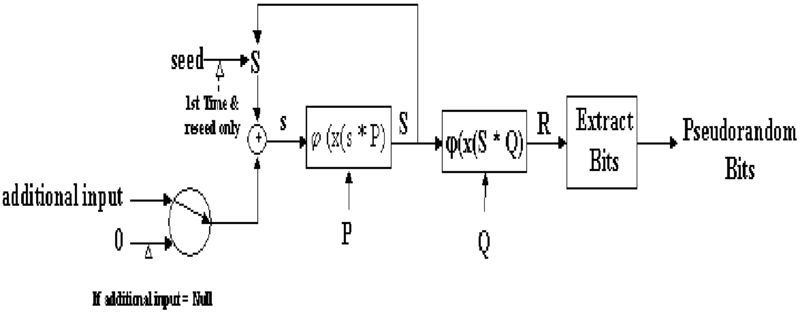
Hannah Davis[1] Matthew Green[2] Nadia Heninger[1]
Keegan Ryan[1] Adam Suhl[1]

[1]UC San Diego

[2]Johns Hopkins University

2004: Dual EC presented at NIST workshop

ECC DRBG Flowchart



July 20, 2004 (dbj)

NIST RNG Workshop
Number Theoretic DRBGs

10

2005-2006: Dual EC standardized in NIST SP 800-90A

A.1 Constants for the Dual_EC_DRBG

The **Dual_EC_DRBG** requires the specifications of an elliptic curve and two points on the elliptic curve. One of the following NIST **approved** curves with associated points **shall** be used in applications requiring certification under [FIPS 140]. More details about these curves may be found in [FIPS 186]. If alternative points are desired, they **shall** be generated as specified in Appendix A.2.

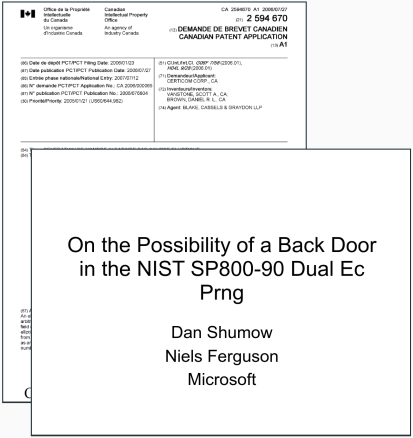
```
Px = 6b17d1f2 e12c4247 f8bce6e5 63a440f2 77037d81 2deb33a0
     f4a13945 d898c296

Py = 4fe342e2 fe1a7f9b 8ee7eb4a 7c0f9e16 2bce3357 6b315ece
     cbb64068 37bf51f5

Qx = c97445f4 5cdef9f0 d3e05e1e 585fc297 235b82b5 be8ff3ef
     ca67c598 52018192

Qy = b28ef557 ba31dfcb dd21ac46 e2a91e3c 304f44cb 87058ada
     2cb81515 1e610046
```

2005-2007: State-recovery backdoor possible in Dual EC


Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada
Canadian Intellectual Property Office
An agency of Industry Canada
CA 2594670 A1 2006/07/27
(21) 2 594 670
(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION
(13) A1
(86) Date de dépôt PCT/PCT Filing Date: 2006/01/23
(87) Date publication PCT/PCT Publication Date: 2006/07/27
(85) Entrée phase nationale/National Entry: 2007/07/12
(86) N° demande PCT/PCT Application No.: CA 2006/000065
(87) N° publication PCT/PCT Publication No.: 2006/076804
(30) Priorité/Priority: 2005/01/21 (US60/644,982)
(51) Cl.Int./Int.Cl. *G06F 7/58* (2006.01), *H04L 9/28* (2006.01)
(71) Demandeur/Applicant: CERTICOM CORP., CA
(72) Inventeurs/Inventors: VANSTONE, SCOTT A., CA; BROWN, DANIEL R. L., CA
(74) Agent: BLAKE, CASSELS & GRAYDON LLP

On the Possibility of a Back Door in the NIST SP800-90 Dual Ec Prng
Dan Shumow
Niels Ferguson
Microsoft

"The relationship between P and Q [in Dual EC] is used as an escrow key and stored. . . the output of the generator [is used] to reconstruct the random number with the escrow key."

2012-2015: Hack of Juniper Network's Dual EC constants

Important Announcement about ScreenOS®

By dscholl posted 12-17-2015 09:02

1 Recommend

IMPORTANT JUNIPER SECURITY ANNOUNCEMENT

CUSTOMER UPDATE: DECEMBER 20, 2015

Administrative Access (CVE-2015-7755) only affects ScreenOS 6.3.0r17 through 6.3.0r20. VPN Decryption (CVE-2015-7756) only affects ScreenOS 6.2.0r15 through 6.2.0r18 and 6.3.0r12 through 6.3.0r20.

We strongly recommend that all customers update their systems and apply these patched releases with the highest priority.

POSTED BY BOB WORRALL, SVP CHIEF INFORMATION OFFICER ON DECEMBER 17, 2015

2012-2015: Hack of Juniper Network's Dual EC constants

Important Announcement about ScreenOS®

By dscholl posted 12-17-2015 09:02

1 Recommend

IMPORTANT JUNIPER SECURITY ANNOUNCEMENT

CUSTOMER UPDATE: DECEMBER 20, 2015

Administrative Access (CVE-2015-7755) only affects ScreenOS 6.3.0r17 through 6.3.0r20. VPN Decryption (CVE-2015-77
through 6.3.0r20.

We strongly recommend that all customers update their systems and apply these patched releases with the highest priority.

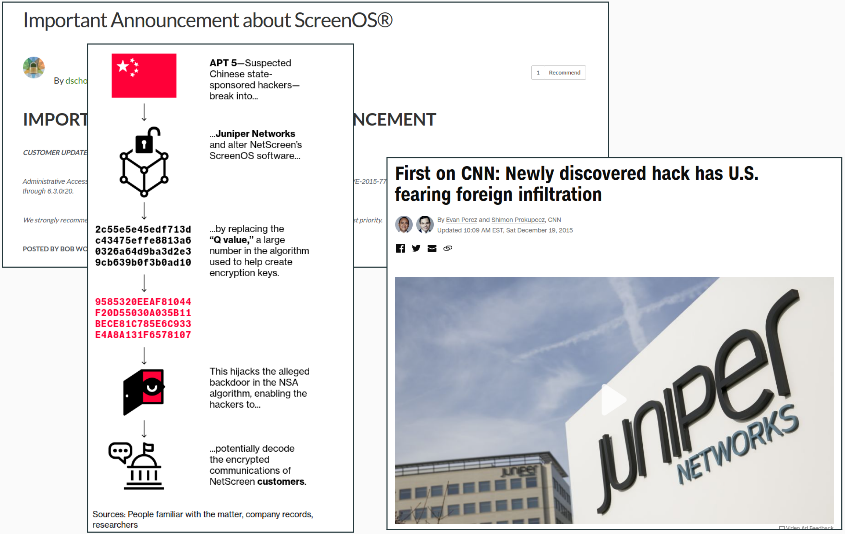
POSTED BY BOB WORRALL, SVP CHIEF INFORMATION OFFICER ON DECEMBER 17, 2015

First on CNN: Newly discovered hack has U.S. fearing foreign infiltration

By Evan Perez and Shimon Prokupecz, CNN
Updated 10:09 AM EST, Sat December 19, 2015

Video Ad Feedback

2012-2015: Hack of Juniper Network's Dual EC constants



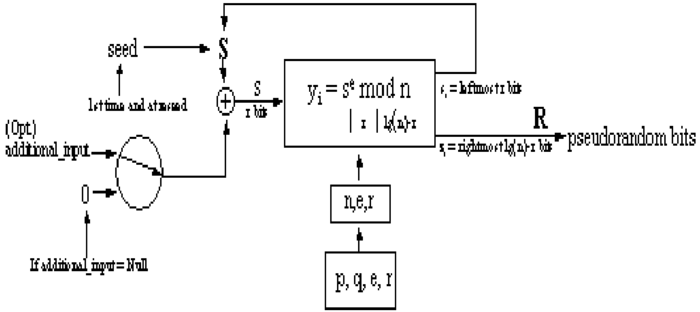
History of Dual EC

Dual EC

- **2004** Proposed inclusion in ANSI x9.82
- **2005** NIST SP 800-9A draft
- **2005-2007** Identification of possible backdoor
- **2013** Snowden Disclosures
- **2014** Removal from SP 800-90A
- **2012-2015** Exploitation of Juniper Networks

2004: Micali-Schnorr presented at NIST workshop

Micali-Schnorr DRBG



July 20, 2004 (dbj)

NIST RNG Workshop
Number Theoretic DRBGs

19

2005: Micali-Schnorr standardized in ISO 18031

Each modulus is of the form $n = pq$ with $p = 2p_1 + 1$, $q = 2q_1 + 1$, where p_1 and q_1 are ($\lg(n)/2 - 1$)-bit primes.

D.2.2 Default modulus *n* of size 1024 bits

The hexadecimal value of the modulus n is:

```
b66fbfda fbac2fd8 2eb13dc4 4fa170ff c9f7c7b5 1d55b214 4cc2257b 29df3f62
b421b158 0753f304 a671ff8b 55dd8abf b53d31ab a0ad742f 21857acf 814af3f1
e126d771 a61eca54 e62bfdb5 85c311b0 58e9cd3f aab758a5 e2896849 6ec1dd51
d0355aa1 55d4d912 6140dcfa b9b03f62 a5032d06 536d8574 0988f384 27f35885
```

D.2.3 Default modulus *n* of size 2048 bits

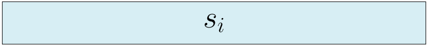
The hexadecimal value of the modulus n is:

```
c11a01f2 5daf396a a927157b af6f504f 78cba324 57b58c6b f7d851af 42385cc7
905b06f4 1f6d47ab 1b3a2c12 17d14d15 070c9da5 24734ada 2fe17a95 e600ae9a
4f8b1a66 96661e40 7d3043ec d1023126 5d8ea0d1 81cf23c6 dd3dec9e b3fce204
5b9299bb cca63dee 435a2251 ad0765d4 9d29db2e f5aba161 279aeb5f 6899fe48
7973e36c 1fb13086 d9231b6b 925a8495 4ba0fbca fea844ea 77a9f852 f86915a4
e71bd0ba b9b269c3 9a7a827a 41311ffa 4470140c 8b6509fe 5dbd39e3 ec816066
2d036e13 0e07e233 06a39b18 db0e8efe 64418880 81ac3673 2b4091f6 63690d03
3b486d74 371a20fc 3e214bce 7ed0e797 5ea44453 cd161d32 e8185204 59896571
```

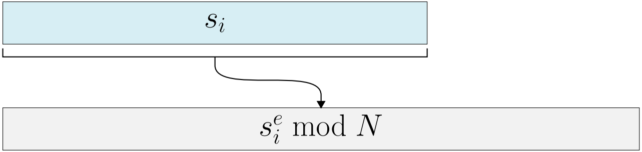
History of Dual EC. . . and Micali-Schnorr

Dual EC		Micali-Schnorr
2004	Proposed inclusion in ANSI x9.82	✓
2005	NIST SP 800-9A draft	ISO 18031
2005-2007	Identification of possible backdoor	?
2013	Snowden Disclosures	
2014	Removal from SP 800-90A	
2012-2015	Exploitation of Juniper Networks	

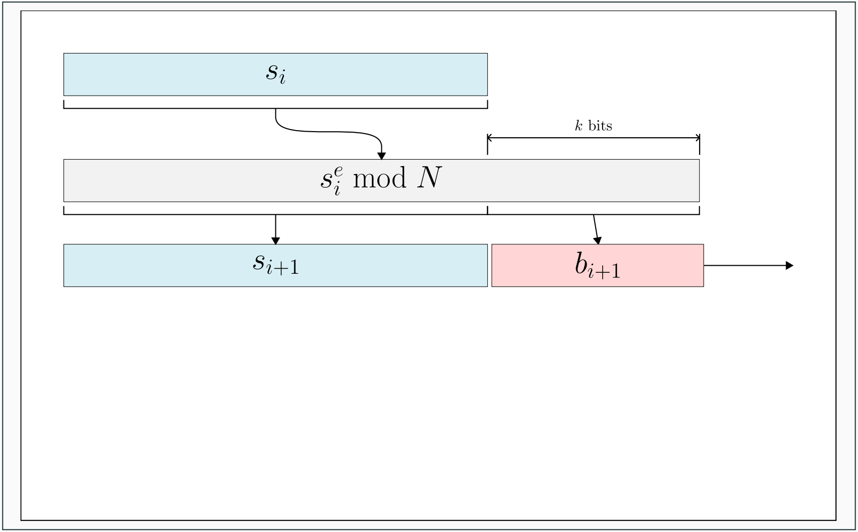
Micali-Schnorr's design: repeated RSA encryption



Micali-Schnorr's design: repeated RSA encryption

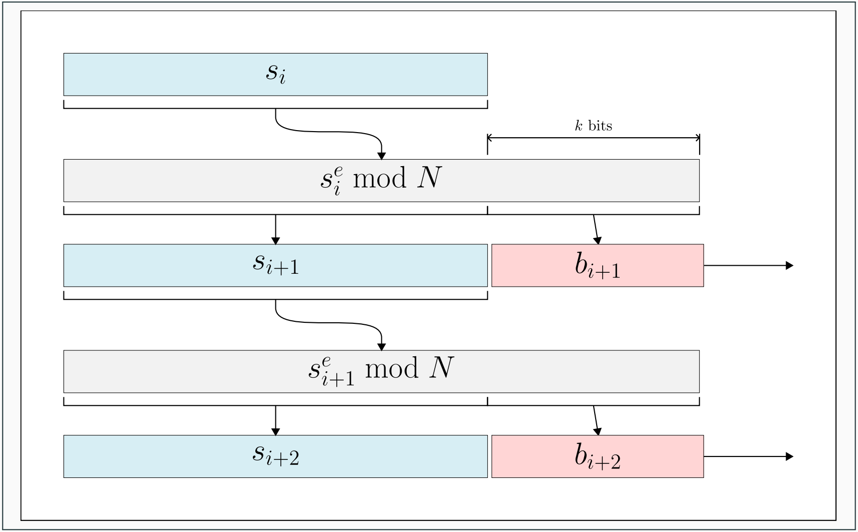


Micali-Schnorr's design: repeated RSA encryption



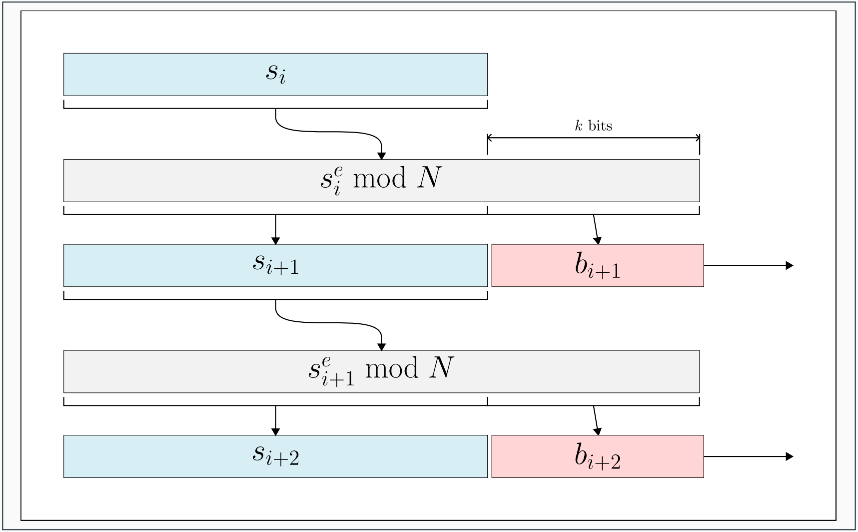
$$2^k s_{i+1} + b_{i+1} \equiv {s_i}^e \pmod{N}$$

Micali-Schnorr's design: repeated RSA encryption



$$2^k s_{i+1} + b_{i+1} \equiv s_i{}^e \pmod N$$

Micali-Schnorr's design: repeated RSA encryption



Unclear how to recover the state using RSA decryption.

Does the factorization of the public modulus lead to an attack against Micali-Schnorr?

Does the factorization, or otherwise malicious construction, of the public modulus lead to an attack against Micali-Schnorr?

Observation 1

There is no *simple* backdoor in Micali-Schnorr.

No simple backdoors in Micali-Schnorr

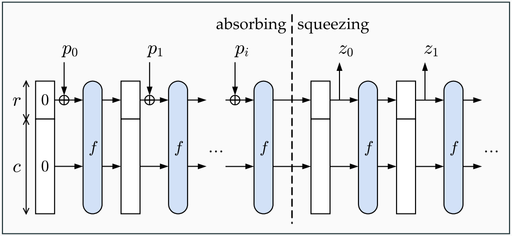
Theorem: Any potential backdoor in Micali-Schnorr must exploit the non-random structure of textbook RSA encryption.

RSA decryption alone is not enough.

No simple backdoors in Micali-Schnorr

Theorem: Any potential backdoor in Micali-Schnorr must exploit the non-random structure of textbook RSA encryption.

RSA decryption alone is not enough.



Micali-Schnorr is like a sponge with duplex construction. It is secure if RSA is replaced with an invertible random function.

Observation 2

There is an algebraic attack on the standard with non-default settings

Algebraic state recovery attacks

We want to recover the unknown state from the observed output.

$$2^k s_{i+1} + b_{i+1} \equiv {s_i}^e \pmod{N}$$

Algebraic state recovery attacks

We want to recover the unknown state from the observed output.

$$2^k s_{i+1} + b_{i+1} \equiv {s_i}^e \pmod{N}$$

Since e is 3 by default, this is a low-degree polynomial with a small solution. Can we use the **multivariate Coppersmith's method**?

Algebraic state recovery attacks

We want to recover the unknown state from the observed output.

$$2^k s_{i+1} + b_{i+1} \equiv s_i{}^e \pmod{N}$$

Since e is 3 by default, this is a low-degree polynomial with a small solution. Can we use the **multivariate Coppersmith's method**?

No. The ISO 18031 state size is not small enough, and this approach fails.

Backdooring Micali-Schnorr with non-default exponent

Backdoor idea: Use non-default public exponent e where the *private exponent* d is small.

Coppersmith's method successfully solves this polynomial.

$$(s_{i+1}2^k + b_{i+1})^d \equiv s_i \pmod{N}$$

Backdooring Micali-Schnorr with non-default exponent

Backdoor idea: Use non-default public exponent e where the *private exponent* d is small.

Coppersmith's method successfully solves this polynomial.

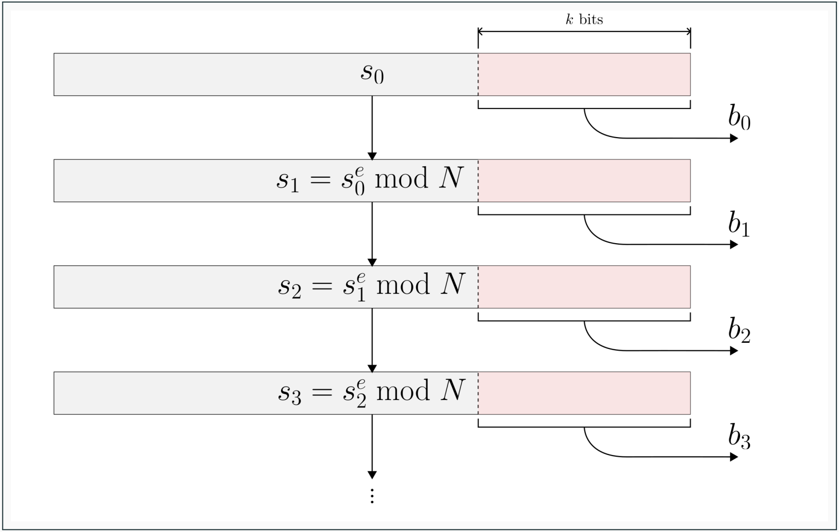
$$(s_{i+1}2^k + b_{i+1})^d \equiv s_i \pmod N$$

ISO 18031: "The implementation should allow" non-default e.

Observation 3

We can force short cycles in a *related* RSA-based construction

RSA PRG



- State $s_i = {s_0}^{e^i} \bmod N$

RSA PRG can have short cycles

RSA PRG with $N = 5154904286740261$ and $e = 3$.

Iteration	Value	State s_i	Output b_i
0	s_0	4047975530247052	`338c`
1	${s_0}^e$	2492861700191393	`34a1`
2	${s_0}^{e^2}$	4862773567328857	`9259`
...	...	...	...
16	${s_0}^{e^{16}}$	810645248255668	`a6b4`
17	${s_0}^{e^{17}}$	2887166220613321	`b6c9`
18	${s_0}^{e^{18}}$	3479941204398616	`d218`

RSA PRG can have short cycles

RSA PRG with $N = 5154904286740261$ and $e = 3$.

Iteration	Value	State s_i	Output b_i
0	s_0	4047975530247052	338c
1	${s_0}^e$	2492861700191393	34a1
2	${s_0}^{e^2}$	4862773567328857	9259
...	...	...	...
16	${s_0}^{e^{16}}$	810645248255668	a6b4
17	${s_0}^{e^{17}}$	2887166220613321	b6c9
18	${s_0}^{e^{18}}$	3479941204398616	d218
19	${s_0}^{e^{19}}$	810645248255668	a6b4

RSA PRG can have short cycles

RSA PRG with $N = 5154904286740261$ and $e = 3$.

Iteration	Value	State s_i	Output b_i
0	s_0	4047975530247052	338c
1	${s_0}^e$	2492861700191393	34a1
2	${s_0}^{e^2}$	4862773567328857	9259
...	...	...	...
16	${s_0}^{e^{16}}$	810645248255668	a6b4
17	${s_0}^{e^{17}}$	2887166220613321	b6c9
18	${s_0}^{e^{18}}$	3479941204398616	d218
19	${s_0}^{e^{19}}$	810645248255668	a6b4
20	${s_0}^{e^{20}}$	2887166220613321	b6c9
...	...	...	...

RSA PRG can have short cycles

- $s_i \equiv s_0^{e^i} \mod N$.
- We're in an exponent in an exponent
- Cycles have length $\varphi(\varphi(N))$
- **Easy to generate parameters where period is very small factor of $\varphi(\varphi(N))$, giving short cycles**
- Such parameters are insecure... but cycling outputs would be obvious.

Observation 4

We can undetectably hide relations between RSA PRG states.

Candidate backdoor for RSA PRG: N embeds sparse relation

Simple relation gives obvious cycles:

$$e^i \equiv e^j \mod \varphi(N)$$
$$\implies s_i \equiv s_j \mod N$$

Cycles (obvious)

But relation with more terms hides cycles:

$$e^h + e^i \equiv e^j + e^\ell \mod \varphi(N)$$
$$\implies s_h \cdot s_i \equiv s_j \cdot s_\ell \mod N$$

No cycles, but still exploitable!

Candidate RSA PRG backdoor:
Choose N to encode a sparse relation between powers of e mod $\varphi(N)$. Exploit via multivariate Coppersmith method.

Unclear how to get backdoor to work for Micali-Schnorr

Truncation prevents us from building exploitable relations

- RSA PRG has an elegant closed form: $s_i = {s_0}^{e^i}$
- MS does not: $s_i = ((((s_0{}^e - b_1)/2^k)^e - b_2)/2^k \ldots$

Conclusion: Need further ideas to extend candidate backdoor to Micali-Schnorr

Recap of our results

- Micali-Schnorr has no "simple" backdoors
    - $\implies$ Any backdoor needs to exploit structure of RSA
- ISO standard allows insecure parameters
- Related construction RSA PRG can be backdoored

Question for the audience

Have you heard of Micali-Schnorr being used in the real world?

If so, please let us know!

Micali-Schnorr: A fun problem that deserves more attention

MS DRBG suspiciously similar to Dual EC DRBG:

- Same origin
- Appear together in ISO 18031
- ISO 18031 specifies default RSA moduli for Micali-Schnorr

But where is the backdoor, if there is one?

- We give partial results and eliminate some avenues of attack
- Question is still open

Questions?

On the Possibility of a Backdoor in the Micali-Schnorr Generator

Full details on ePrint: `https://ia.cr/2023/440`