# Revisiting Discrete-Log Based Random Number Generators (or: How to Fix EC-DRBG?)

–

René Struik

(Struik Security Consultancy)

e-mail: rstruik.ext@gmail.com

# Outline

1. Notation
2. NIST EC-DRBG
   – Description of Generator
   – Security Caveats
3. EC-DRBG “Fixes”
   – Main Objectives
   – Five Constructions
4. Conclusions

# Notation

$E(\mathbf{F}_q)$: elliptic curve over field $\mathbf{F}_q$
**G**: cyclic subgroup of $E(\mathbf{F}_q)$, of prime-order $n$
$G$: base point of **G**
$h$: co-factor (usually, small)

One has $|E(\mathbf{F}_q)| = n \cdot h$

$x(P)$: $x$-coordinate of point $P$ on the curve (not being point at infinity), when represented in affine coordinates

# NIST EC-DRBG Generator

**Algorithm 1: EC-DRBG Generator**

**Input:** $k \in \mathbf{Z}_q$, $b \leq q$, $l \geq 0$

**Output:** $l$ pseudorandom numbers in $\mathbf{Z}_b$

**for** $i$:=1 **to** $l$ **do**

Set $(R, S) \leftarrow (kG, kQ)$;

Set $(k, out_i) \leftarrow (x(R)$ (**mod** $q$), $x(S)$ (**mod** $b$));

**end for**

Return $(out_1, \ldots, out_l)$

NIST EC-DRBG:

− $b$: a power of two (i.e., output obtained via truncation of $x$-coordinate)

− $b$: at least 13 + $\log_2 h$ bits less than bit-size of order of finite field $\mathbf{F}_q$ (byte-oriented)
(recommendation was to pick $b$ as large as possible, for efficiency reasons)

− $E(\mathbf{F}_q)$: NIST prime curves P-256, P-384 (and others)

− $G$, $Q$: default values specified for NIST prime curves P-256, P-384
(alternative values allowed, provided generated *verifiably at random*)

# Security of NIST EC-DRBG

1. Potential back-door EC-DRBG

Unknown whether default base point $G$ and public key $Q$ generated verifiably at random

Unknown if $\log_G(Q)$ known to those who specified $G$ and $Q$

- If $d := \log_G(Q)$, one can determine internal state $R$ from $S$, since $R := d^{-1}S$
- One can determine $S$ from $x(S)$, since only two points with same $x$-coordinate
- One can determine $x(S)$ from truncated version, since only roughly 16 bits removed

So, if $\log_G(Q)$ known, then internal state leaked from observed output $out_i$

2. Output EC-DRBG distinguishable from random bit string

- Set of $x$-coordinates of valid point forms subset of $\mathbf{F}_q$ of cardinality roughly $q/2$ and easy to check whether $x \in \mathbf{F}_q$ is in this set. So, output of EC-DRBG (without truncation) is easily distinguished from random element of $\mathbf{F}_q$
- Distinguishability remains with truncation, if one does not remove sufficiently many bits from $x(S)$

3. Loose security reduction

Hardness of so-called $x$-Logarithm Problem, on which security of core EC-DRBG relies, is hard to quantify and security reduction of related security problem (AXLP) to Diffie-Hellmann problem (DDH) is rather loose

# NIST EC-DRBG "Fixes"

Minor "tweaks" of EC-DRBG suffice to obtain the following properties:

1. Reduce/remove reliance on public key $Q$
2. Lower distinguishability of output bit string
3. Tighten security reductions
4. Provide potential resilience against quantum cryptographic attacks (should these become a long-term threat)

**Claims:**

− Techniques apply to short Weierstrass curves (e.g., NIST, Brainpool), Montgomery curves, Edwards and twisted Edwards curves, binary curves.
− Techniques do not add additional computational cost (mostly, far more efficient)
− Techniques can do without public key $Q$, thus eliminating key substitution attacks

**NOTE:** builds upon existing cryptanalysis EC-DRBG ([1])
− uses tight bounds on character sums and Kloosterman sums ([18])
− uses presumed difficulty of Diffie-Hellman problems ([7])

# Example of 'Fix' (roughly "Construction C")

**Original EC-DRBG Generator**

**Input:** $k \in \mathbf{Z}_q$, $b \leq q$, $l \geq 0$

**Output:** $l$ pseudorandom numbers in $\mathbf{Z}_b$

**for** $i$:=1 **to** $l$ **do**

Set $(R, S) \leftarrow (kG, kQ)$;

Set $(k, out_i) \leftarrow (x(R)$ (**mod** $q$), $x(S)$ (**mod** $b$));

**end for**

Return $(out_1, \ldots, out_l)$

"Algorithm C": DDH Generator

**Input:** $k \in \mathbf{Z}_q$, $l \geq 0$

**Output:** $l$ pseudorandom numbers in $\mathbf{Z}_b$

**for** $i$:=1 **to** $l$ **do**

Set $(R, S) \leftarrow (kG, kQ)$;

Set $(k, out_i) \leftarrow (x(R)$ (**mod** $q$), $(x(R) + x(\mathrm{S}))$ (**mod** $b$);

**end for**

Return $(out_1, \ldots, out_l)$

# NIST EC-DRBG vs. New DDH Constructions

| **Construction** | **NIST** | **A** | **B** | **C** | **D** | **E** | **D(*k*)** |
|---|---|---|---|---|---|---|---|
| *#Public keys **Q*** | 1 | 3 | 2 | 1 | − | − | − |
| *≈ # rnd. bits/curve size* | 1 | 1 | 1 | 1 | 1 | 1 | *k* |
| *Rate*[1] | 1/2 | 1/4 | 1/3 | 1/2 | 1/3 | 1/2 | *k*/(*k*+2) |
| *Backdoor possible?* | *Yes* | *unlikely* | *unlikely* | *unlikely* | *No* | *No* | *No* |
| *Indistinguishable output* | *poor* | | | | | | |
| *- if state **R** not known* | | *tight* | *tight* | *tight* | *tight* | *tight* | *tight* |
| *- if state **R** known* | | *tight* | *tight* | *poor* | *tight* | *poor* | *tight* |
| *Reduction next state* | *AXLP* | | | | | | |
| *- if output not known* | | *tight* | *tight* | *tight* | *tight* | *tight* | *tight* |
| *- if output known* | | *tight* | *tight* | *AXLP* | *tight* | *AXLP* | *tight* |
| *Quantum-crypto secure?* | *No* | *perhaps* | *perhaps* | *perhaps* | *likely* | *likely* | *likely* |

**Notes:**

- Five constructions submitted to NIST (as comment re-opened SP 800-90A spec)
- Full details in draft technical paper

[1]Rate: #random bits (as multiple of bit-size curve)/#scalar multiplications

# Conclusions

Security weaknesses EC-DRBG relatively easy to fix

- Five constructions, with slightly differing properties
- Simplest fix: only change w.r.t. original EC-DRBG is *single modular addition*
- Some suggested fixes possibly resistant to quantum-cryptographic attacks

Constructions work for “short” Weierstrass curves (e.g., NIST, Brainpool), Edwards curves, twisted Edwards curves, Montgomery curves

| |
|---|
| Contrary to popular belief, NIST EC-DRBG can be made highly secure |

**Notes:**

- Main constructions submitted to NIST
- Full details to appear in technical paper

# Further Reading

1. D.R.L. Brown, K. Gjøsteen, "A Security Analysis of the NIST SP 800-90 Elliptic Curve Random Number Generator," in *Proceedings of Advances of Cryptology – CRYPTO 2007*, A. Menezes, Ed., Lecture Notes in Computer Science, Vol. 4622, pp. 466-481, Berlin: Springer, 2007.
2. R.R. Farashahi, B. Schoenmakers, A. Sidorenko, "Efficient Pseudorandom Generators Based on the DDH Assumption," in *Proceedings of 10th International Conference on Practice and Theory in Public-Key Cryptography – PKC 2007*, T. Okamoto, X. Wang, Eds., Lecture Notes in Computer Science, Vol. 4450, pp. 426-441, Berlin: Springer, 2007.
3. B. Schoenmakes, A. Sidorenko, "Cryptanalysis of the Dual Elliptic Curve Pseudorandom Generator," IACR ePrint 2006-190.
4. NIST SP 800-90, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, National Institute of Standards and Technology, Department of Commerce, 2007.
5. ANS X9.82: Part 3-2007, *Random Number Generation, Part 3: Deterministic Random Bit Generators*, American National Standard for Financial Services, Accredited Standards Committee X9, Annapolis, MD, 2007.
6. D. Brown, R. Gallant, "The static Diffie-Hellman Problem," International Association for Cryptologic Research, IACR ePrint 2004-306.
7. N. Koblitz, A. Menezes, "Intractable Problems in Cryptography," in *Finite Fields: Theory and Applications*, Contemporary Mathematics, 518, pp. 279-300, 2010.
8. P-A. Fouque, A. Joye, M. Tibouchi, "Injective Encodings to Elliptic Curves," in *18th Australasian Conference on Information Security and Privacy – ACISP-2013*, C. Boyd, L. Simpson, Eds., Lecture Notes in Computer Science, Vol. 7959, pp. 203-218, New York: Springer, 2013.

# Further Reading (cont'd)


9. D.J. Bernstein, M. Hamburg, A. Krasnova, T. Lange, “Elligator: Eliptic-Curve Points Indistinguishable from Uniform Random Strings,” in *Proceedings of 20th ACM Conference on Computer and Communications Security – ACM-CCS 2013*, 2013.
10. D.F. Aranha, P.S.L.M. Barreto, G.C.C.F. Pereira, J.E. Ricardini, “A Note on High-Security General-Purpose Elliptic Curves,” International Association for Cryptologic Research, IACR ePrint 2013-647.
11. J.H. Cheon, “Discrete Logarithm Problems with Auxiliary Inputs,” J. of Cryptology, 2010.
12. D. Shumow, N. Ferguson, “On the Possibility of a Backdoor in the NIST SP 800-90 Dual-ECC PRNG,” Rump Session, Crypto 2007.
13. FIPS Pub 186-2, *Digital Signature Standard (DSS)*, Federal Information Processing Standards Publication 186-2, US Department of Commerce/National Institute of Standards and Technology, Springfield, Virginia, January 27, 2000. (Includes Change Notice, October 5, 2001.) \
14. NIST SP 800-90A, *Recommendation for Random Number Generation Using the Deterministic Random Bit Generators*, Revision 1, Draft, National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD, September 9, 2013.
15. NIST SP 800-90B, *Recommendation for the Entropy Sources Used for Random Bit Generation*, Draft, National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD, September 6, 2012.
16. NIST SP 800-90C, *Recommendation for Random Bit Generation Constructions*, Draft, National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD, September 6, 2012.

# Further Reading (cont'd)


17. D.R. Hankerson, A.J. Menezes, S.A. Vanstone, *Guide to Elliptic Curve Cryptography*, New York: Springer, 2003.
18. R. Lidl, H. Niederreiter, *Finite Fields*, 2nd Edition, Cambridge: Cambridge University Press, 1997.
19. BSI , *Technical Guidance TR-03111 – Elliptic Curve Cryptography, Version 2.0*, June 28, Bundesamt fur Sicherheit in der Informationstechnik, Bonn, Germany, 2012.
20. RFC 5639, Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, March 2010.
21. L.K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search," STOC 1996.