

JEnt v3.7.0 XDRBG Conditioning Validation

(updated to reflect some of the CMUF EWG meeting discussion)

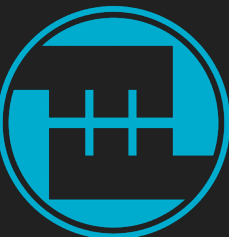
Joshua E. Hill, PhD



20260708
CMUF EWG

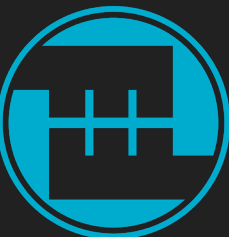
Introduction

- This attempts to summarize the discussion in JEnt's GitHub, [Issue #213](#).
- Generally, there have been two evaluation issues raised with use of the XDRBG construction.
- These will (mostly) be resolved if XDRBG becomes an approved DRBG (via IG D.K Resolutions 5 and 7).
- Before XDRBG is an approved DRBG, validations will likely decompose the XDRBG into a chain of two (vetted) SHAKE256 ioning stages:
 1. Reseed
 2. Generate



SHAKE256 XDRBG, Reduced

- Under the convention that the j th XDRBG state is V_j , and the i th “intermediary” block containing the raw delta value is X_i , the SHAKE256 XDRBG reduces to something like (**wave hands**: this disregards parameter ordering, and the deterministic encoding step):
 1. Reseed: $V_t = \text{SHAKE256}(512, V_{t-1} || X_1 || \dots || X_N)$
 2. Generate: $(V_{t+1}, S) = \text{SHAKE256}(768, V_t)$



Issue #1: The Maximum SHAKE256 Output

- The [List of Vetted Conditioning Components](#) limits SHAKE256 output (n_{out}) to 512 bits.
- Chris Celi (NIST) noted that a similar restriction exists for all the listed vetted conditioning components.
 - n_{out} was limited to at most the algorithm's narrowest width, thus preventing the conditioning functions from expanding data.
 - This restriction is in tension with the expected use of various XOF functions.



Issue #1: The Maximum SHAKE256 Output

- XDRBG Generate causes the XOF to produce a string the length of the (next) XDRBG state + the size of the Generated XDRBG output.
- [The XDRBG specification](#) suggests that the SHAKE256-based XDRBG construction should have a state size of 512 bits, a maxout (maximum output length of Generate) of 2752 bits, and a targeted security strength of 256 bits.
 - In the general SHAKE256 XDRBG Generate, SHAKE256 may produce as many as $2752+512=3264$ bits.
- JEnt's XDRBG Generate produces a fixed 256-bit long output, so here the SHAKE256 function always produces $256+512=768$ bits.
- When validated DRBGs are used as conditioning components, IG D.K Resolution 5 limits the Generate output length to at most the security strength of the DRBG.
 - The JEnt behavior is consistent with this requirement.

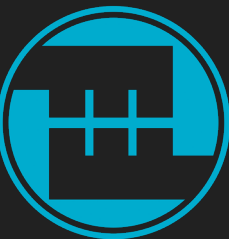


Issue #2: What is “ n_{out} ” for the Generate Stage?

- JEnt’s XDRBG Generate (step 2):

$$(V_{t+1}, S) = \text{SHAKE256}(768, V_t)$$

- If the full result of SHAKE256 is considered “output”, then any XDRBG Generate call produces more than the n_{out} limit.
- The JEnt XDRBG Generate takes in roughly 321 bits of entropy and SHAKE256 produces a 768-bit string.
 - This string is partitioned between the 512-bit next state (V_{t+1}) and the 256-bit output (S).
 - This string contains about 321 bits of entropy.



Issue #2: What is “ n_{out} ” for the Generate Stage?

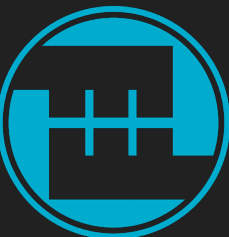
- In JEnt, the XDRBG state (V_j) is never output from the entropy source.
 - The next state output from the Generate function is only used in the next Reseed operation.
- In JEnt, only the 256-bit S is output from the entropy source.
 - This is the conditioned entropy from the entropy source.
- Question:

For JEnt v3.7.0, should we treat the Generate SHAKE256 stage n_{out} as 768 bits, or as 256 bits?



Issue #2: What is “ n_{out} ” for the Generate Stage?

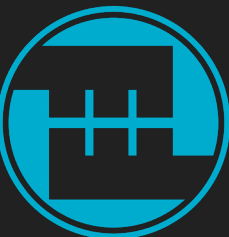
- If we consider the output of the SHAKE256 to be only S , then $n_{\text{out}} = 256$.
- Else, (assuming that Issue #1 is resolved) if we instead consider the output of SHAKE256 to be the 768-bit output of a vetted conditioning function and...
 - If S is considered the 256-bit truncation of this full string, then IG D.K Resolution 20 directs us to set $n_{\text{out}} = 256$.
 - Else we can only credit a linear scaling of the min entropy present in the full 768-bit output string.



Issue #2: What is “ n_{out} ” for the Generate Stage?

Some Notes:

- We can't credit the feedback to the Reseed operation (V_{t+1}) as having entropy (by IG D.K Resolution 6).
 - None of the contemplated answers to this question results in “double counting” entropy.
- In any case, if $n_{\text{out}} = 256$ for JEnt's XDRBG Generate, then IG D.K Resolution 19 allows for a “full entropy” claim for the produced S .
- If we have to linearly scale the entropy (as per SP 800-90B Section 3.1.5.1.2), then the produced S would have $\approx 321 \times \frac{256}{768} = 107$ bits of min entropy.



Final Notes and a Proposed Resolution

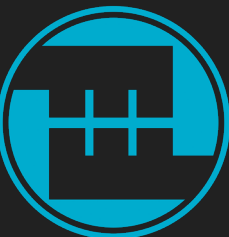
- The security characteristics of the JEnt v3.7.0 conditioning component are excellent.
- n_{out} ought to be understood as the size of the data being output forward in the conditioning chain.
 - In this case this is the size of the data output from the entropy source, as the Generate SHAKE256 is the terminal conditioning stage in the conditioning chain.
- This would:
 - Resolve the XDRBG SHAKE256 Generate n_{out} problem.
 - Allow a post-conditioned "full entropy" claim for JEnt v3.7.0.



Final Notes and a Proposed Resolution

What is the best way for CMVP to accomplish this? That's not clear, but here is one proposal:

- Revise IG D.K Resolution 4 to add bullet c covering the parameter n_{out} . Proposed text:
 - c. The parameter n_{out} , the amount of data output from the conditioning component out of the entropy source (for the final conditioning component in the chain) or to the next conditioning component in the chain (for all other conditioning components).



References

- Chris Bell, et al. [Jitter Entropy Library GitHub Issue #213](#). Accessed July 7, 2026.
- John Kelsey, Stefan Lucks, and Stephan Müller. [XDRBG: A Proposed Deterministic Random Bit Generator Based on Any XOF](#). IACR Transactions on Symmetric Cryptology, 2024(1), 5-34.
- Stephan Müller. [CPU Time Jitter Based Non-Physical True Random Number Generator](#). May 4, 2026.
- CMVP. [Updates on Recently Approved Algorithms in Random Bit Generation Standards](#). Updated April 13, 2026.
- CMVP. [Implementation Guidance for FIPS 140-3 and the Cryptographic Module Validation Program](#). April 16, 2026.

