One Operation, Polynomial Forwards and Exponential Backwards: But a Large Ratio Is Not Enough

— A key exchange was carried out for the modulus $2^{127} - 1$ and both sides were confirmed to reach the same shared secret — the separator is whether the inverse problem is easy or hard — [Paper 676]

Yuuki YAMAGISHI (山岸 勇輝)
Yamagishi Complexity Torus Lab, Japan
September 2026

— *From thinking to syncing.* —

Abstract

Two people can build a secret known only to them by exchanging numbers in the open, because one operation is easy forwards and hard backwards.That ease and that hardness are counted, and how large the ratio must be before one may call it safe is examined.No new theorem or law is claimed. First, the procedure really runs——a key exchange for the modulus $2^{127} - 1$ was carried out, and both sides, starting from different secrets, reached the same shared secret (Section 2). **Second, the two directions differ by orders——making it takes about 254 multiplications and breaking it, by a square-root attack, about 1.304382×10^{19} operations, a ratio of 5.135361×10^{16}** (Section 2). **Third, and this is the core. A large ratio is not enough——even with that ratio of 5.135361×10^{16}, a machine doing a trillion operations a second breaks the key in 0.413334 years** (Section 3). **Fourth, doubling the bits acts differently on the two sides——the cost of making it rises only 2.000000 times while the cost of breaking it rises 1.304382×10^{19} times, putting the time to break at 5.391449×10^{18} years** (Section 3). **Fifth, so a key length is chosen by absolute size, not by ratio——the ratio takes the same form at 127 bits and at 254, while the time to break divides into 0.413334 years and 5.391449×10^{18} years** (Section 3). **Sixth, the separator is whether the inverse problem is easy or hard——the wager is on hardness, not on impossibility** (Section 4). **Keywords**: key exchange, discrete logarithm, one-way function, computational cost, public-key cryptography, key length.

Scope of this paper (scope note): **No new theorem or law is claimed——key exchange in the open (Diffie and Hellman 1976), the hardness of the discrete logarithm, and square-root attacks are all standard (Menezes, van Oorschot and Vanstone 1996).No scheme is evaluated——which group to use, and whether to use elliptic curves or finite fields, are not the subject, and no product or standard is named.Authentication is not treated——the man-in-the-middle attack and the authentication that prevents it are not the subject. Only the numbers of the exchange itself are examined.Quantum computers are not treated——quantum speed-ups belong to Paper 305; only classical costs are counted here.Implementation is not discussed——the quality of randomness, side channels and implementation errors are not the subject.The cost estimates are representative——taking the cost of making it as multiplications proportional to the number of bits, and of breaking it as the square root of the group size, is standard; real attacks may be faster or slower.Relation to earlier papers**: **Paper 305 showed that "security" rests on three grounds and that a quantum computer breaks only one——this paper enters the computational ground and counts both directions of one procedure.Paper 304 showed that "capacity" is a number attainable only at infinite length——there "attained" and "approached" divided; here "unbreakable" and "slow to break" do.**What is added is **computing the shared secret and confirming that both sides agree**, **putting a number on the ratio of the two directions**, **showing that at 127 bits the key still falls in 0.413334 years**, **setting the two sides' responses to doubling the bits side by side**, and **placing the separator on the inverse problem**.

1. Notation and the Question

Setting: **a prime p and a generator g are public. Each party chooses a secret number, a or b, and publishes g^a or g^b. The shared secret agrees because**

$$(g^b)^a \bmod p = (g^a)^b \bmod p$$

Making it is repeated squaring, with a number of multiplications proportional to the number of bits. Breaking it requires recovering a from g^a, the discrete logarithm.

Question: **what is security wagering on: impossibility, or a difference in cost?**

2. Counting Both Directions

Computation 2.1: **a key exchange was carried out with the modulus $2^{127} - 1$. The two sides, computing from different secrets, reached the same shared secret.**

Computation 2.2: **the costs were counted.**

bits in the modulus	making it (multiplications)	breaking it (operations)
127	254	1.304382×10^{19}
254	508	1.701412×10^{38}
508	1016	2.894802×10^{76}

At 127 bits, breaking costs 5.135361×10^{16} times as much as making.

> **Observation 2.3**
>
> **two directions of one operation, orders apart.Raising to a power is work proportional to the number of bits; extracting the exponent again is work proportional to the square root of the size of the group——polynomial and exponential live inside a single operation.**

3. A Large Ratio Is Not Enough

Computation 3.1: **the time to break was derived for a machine trying a trillion operations a second.**

bits in the modulus	time to break
127	0.413334 years
254	5.391449×10^{18} years
508	9.173075×10^{56} years

> **Observation 3.2**
>
> **this is the core: the same ratio, different answers.At 127 bits the ratio is still 5.135361×10^{16} and the shape, easy to make and hard to break, still holds. And the key falls in 0.413334 years——to be safe, the ratio must be large and the absolute number of operations on the breaking side must be large too.**

Computation 3.3: **doubling the number of bits raises the cost of making by 2.000000 and the cost of breaking by 1.304382×10^{19}.**

Observation 3.4

here is where the defender has the advantage.The cost of adding bits is linear, and the cost of attacking grows exponentially——so one and the same move, “make it a little longer”, means something of a different order on each side.

4. The Separator

Observation 4.1

the separator is whether the inverse problem is easy or hard.What was counted is cost, not impossibility. Find one better way to invert and the ratio shrinks by that much——security wagers on hardness, and the size of the wager can be said in numbers.

Observation 4.2

beside Paper 305.Paper 305 divided security into three grounds and placed, as the strongest, that what carries no information cannot be computed by any machine——this paper counts the neighbouring ground, the one that wagers on computation. The figure 0.413334 years shows that the wager depends on the choice of size.

Observation 4.3

beside Paper 304.Paper 304 showed that “capacity” is attainable only at infinite length——there “attained” and “approached” divided; here “unbreakable” and “slow to break” do. Both are steps between a word of the limit and finite practice.

Observation 4.4

what is not said.Neither the man-in-the-middle attack, nor the choice of group, nor quantum computers is treated, and the cost of breaking is a standard estimate——what is shown is the difference in cost between the two directions of one operation, and that the difference alone does not settle safety.

5. Roll Call and Closing

1. **both sides reach one secret** (Computation 2.1): **the procedure runs.**
2. **254 multiplications to make** (Computation 2.2): **proportional to bits.**
3. **1.304382×10^{19} to break** (Computation 2.2): **the square-root cost.**
4. **a ratio of 5.135361×10^{16}** (Computation 2.2): **polynomial beside exponential.**
5. **127 bits fall in 0.413334 years** (Computation 3.1): **the core.**
6. **254 bits take 5.391449×10^{18} years** (Computation 3.1): **same ratio, other answer.**
7. **508 bits, 9.173075×10^{56} years** (Computation 3.1): **the effect of size.**
8. **making costs 2.000000 times** (Computation 3.3): **linear.**
9. **breaking costs 1.304382×10^{19} times** (Computation 3.3): **exponential.**
10. **easy or hard** (Observation 4.1): **the separator; Papers 305 and 304.**

Closing: **Key exchange works because one operation is polynomial forwards and exponential backwards.For the modulus $2^{127} - 1$ the shared secret was confirmed to agree, and the costs counted: 254 multiplications to make against 1.304382×10^{19} operations to break, a ratio of 5.135361×10^{16}.And still a machine doing a trillion operations a second breaks it in 0.413334 years——a large ratio is necessary and not sufficient; what is needed is absolute size on the breaking side.Doubling the bits leaves the cost of making at 2.000000 times and puts the time to break at 5.391449×10^{18} years——one move, two orders of meaning.The separator is whether the inverse problem is easy or hard.Placed among the earlier papers——of the three grounds Paper 305 divided, the computational one was counted here; as in Paper 304, the step lies between a word of the limit and finite practice.To be honest——the cost of breaking is a standard estimate, and a faster method would shrink the ratio.**

References

1. Diffie, W., & Hellman, M. (1976). New Directions in Cryptography. IEEE Transactions on Information Theory.
2. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). *Handbook of Applied Cryptography*. CRC Press.
3. Yamagishi, Y. (2026). *"Capacity" Is a Number Attainable Only at Infinite Length* [Paper 304]. Zenodo. 10.5281/zenodo.22172081
4. Yamagishi, Y. (2026). *Security Rests on Three Grounds, and a Quantum Computer Breaks Only One* [Paper 305]. Zenodo. 10.5281/zenodo.22172118

— From thinking to syncing. —

Yuuki YAMAGISHI (山岸 勇輝)

© 2026 Yamagishi Complexity Torus Lab, Japan · Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0)