Layering diverse cryptography to lower future secret attack risks: post-quantum estimates

Daniel R. L. Brown*

May 10, 2021

Abstract

Layering diverse cryptography is a general method to lower the risk of a future, or secret, cryptanalytic attack on a system. This report describes methods to quantifiably estimate this risk reduction.

Diversity is especially helpful in forward security because future attackers have more time to discover new attacks, making attack independence of diverse cryptography the major contribution to risk reduction. Post-quantum security is a part of forward security.

Estimates for highly sensitive data say that the security advantage of diverse layering is worth the extra usage cost, thus advising a decision to layer diverse cryptography.

1 Introduction

The goal of post-quantum cryptography (PQC) is to hedge the risk that a quantum computer might break ECC (or RSA) by running Shor's algorithm. An attacker who hides the existence of its Shor-running quantum computer can run a secret attack against ECC (or RSA) users. So, PQC hedges this type of secret attack.

Layering diverse cryptography is also a method for hedging the risk of secret attacks (more general secret attacks, not just quantum attacks). This alignment in purpose suggests that PQC and layering diverse cryptography share a common purpose, and should perhaps be used in tandem.

*danibrown@blackberry.com

2 Definitions

2.1 Secret and public attacks

An **attack** against a cryptographic scheme is a feasible method to defeat the stated security aim of the scheme. An attack is a **public attack** if reasonable verification of the attack is available to the general public. Otherwise, an attack is a **secret attack**.[1] The general public can, at best, estimate the risk of a secret attack.[2]

For example, a feasible method to defeat the security aims of Elliptic Curve Diffie–Hellman (ECDH) would be Shor's algorithm with a large enough quantum computer. This would be a public attack, if the existence of a large enough quantum computer can be verified by the general public, or possibly if a break of ECDH is demonstrated (such as by a solution to one of the larger Certicom ECC challenges). Otherwise, it should be considered a secret attack.

It is important to consider that some cryptographic schemes have **future security** aims: meaning that they try to protect today's data from **future attacks**, attacks discovered in the future. When future security is an aim, such as in encryption, potential future attacks are counted as secret attacks, whether or not future attacks are made public. When future security is not an aim, such as in authentication, then future attacks are not counted at all.

2.2 Strongest-link layering

Given a suite of cryptographic schemes $[C_1, \ldots, C_n]$, all with the same security aim, such as

- four key encapsulation schemes: [ECDH, NTRU, McEliece, SIKE], or
- three signature schemes: [ECDSA, Dilithium, SPHINCS+],

a **strongest-link layering** is a scheme written as

$$C = C_1 \,\&\, C_2 \,\&\, \ldots \,\&\, C_n, \tag{1}$$

with the same security aim as the C_i, such that an ability to break C implies an ability to breaking each C_i individually, and conversely, an ability to break

[1] A secret attack is a type of **zero-day vulnerability**.

[2] Unless investigators or whistle-blowers make the secret attack public.

all of the C_i separately implies an ability to break C. In other words, as long as one or more of the C_i is unbroken, the layered scheme C is unbroken.

This report assumes that such strongest-link layering is possible, and estimates the benefits of strongest-link layering.

2.3 Attack probability

Let E indicate the **event of a secret attack** against scheme C, and E_i indicate the event of a secret attack against scheme C_i. If $C = C_1 \,\&\, \ldots C_n$, then $E = E_1 \cap \cdots \cap E_n$, the intersection of the events E_i, because of the definition of strongest-link layering.[3]

The **(secret) attack probability** a against C, is the probability of the event E of a secret attack against C, which can be written as $a = P(E)$. Similarly, $a_i = P(E_i)$ is the attack probability against C_i. If C is a strongest-link layering scheme, then $a = P(E_1 \cap \cdots \cap E_n)$.

2.4 Expected loss (risk)

The risk of secret attacks depends both on the probability a of secret attack and on the **damage** D that would be caused by a secret attack.

The **damage** D of an attack breaking security aims of scheme C depends on the application[4] using the scheme C and the type of data protected by the scheme C. This report treats D as a given and unchangeable single financial number, in units of dollars.[5]

The risk of secret attacks is equated in this report to **expected loss** L defined as

$$L = aD, \tag{2}$$

where a is the probability of a secret attack and D is the damage that would result from that secret attack.

2.5 Usage cost

The **usage cost** U of scheme C is the cost of using C, and should cover computer runtime, data transmission, software (or hardware) installation.[6]

[3] We assume that there are no (current) public attacks on any of E_i.

[4] Such as email or web browsing.

[5] The value of damage may also depend on the identity of the secret attacker.

[6] The usage cost does not include risk of costs caused by attackers.

Similarly, let U_i be the usage cost for scheme C_i. This report treats U and U_i as given, unchangeable financial numbers, in units of dollars.

A strongest-link layering $C = C_1 \,\&\, \ldots \,\&\, C_n$ typically has **additive usage** cost of

$$U = U_1 + \cdots + U_n. \tag{3}$$

2.6 Net cost, benefit and net benefit

The **net cost** N of a cryptographic scheme C is the usage cost plus the expected loss:

$$N = U + L. \tag{4}$$

The **benefit** B of using the scheme C depends on the application using the scheme C and the type of data protected by the scheme C. This report treats B as a given and unchangeable single financial number, in units of dollars.

The **net benefit** is

$$B - N, \tag{5}$$

the benefit minus net cost. We need a positive net benefit ($B - N > 0$).

If the net benefit is not positive ($B - N \leq 0$), then the risk of a secret attack against the scheme C is too high. This would indicate that better cryptography is needed, or else something beyond cryptography, such as in-person, physical, communication.

The requirement $B - N > 0$ means requiring that $N < B$. Therefore B can viewed as a maximum threshold for N. Cryptographers must try to reduce N such that N is below B. Furthermore, cryptographers also want to maximize the net benefit $B - N$ by minimizing the net cost N (among all acceptable options $N < B$).

3 Estimates

3.1 Thoughtover estimates for a

The **thoughtover estimate** $\tilde{a}$ for secret attack probability a of a cryptographic scheme C is

$$\tilde{a} = 1 - o^{t/T} \tag{6}$$

where:

- T is independent **public thought** put towards attacking C, the total time spent thinking how to break the scheme by those who would publish their attacks if discovered;
- t is independent **secret thought** put towards attacking C, the total time spent thinking how to break the scheme by secret attackers, who would not publish their attacks if discovered; if a scheme C has an aim to provide future security, its t should also include the potential time the all relevant future attackers would spend thinking of how break the scheme;
- o is **optimism** (or confidence or P-value or statistical significance).

See §4 for justification and discussion of the thoughtover estimate.

A thoughtover estimate $\widetilde{a_i}$ for a_i is defined similarly, as

$$\widetilde{a_i} = 1 - o^{t_i/T_i}. \tag{7}$$

3.2 Cautious optimism

Fixing $o = 0.05$ is **cautious optimism**. Cautious optimism is derived from the typical cut-off for **statistical significance** of 95% used commonly in many sciences. If $t = T$, the public cryptanalysis and secret cryptanalysis should have equal chances of finding an attack. Putting $o = 0.05$, allows for a 95% probability that secret attackers succeed while public attackers fail. In other words, putting $o = 0.05$ accounts for public attackers having worse luck than the secret attackers.

Putting $o = 0.5$ would not account for the possibility of the secret attackers being luckier at finding attacks than the public attackers.

Any $o > 0.5$ is over-confidence, assuming the public attackers have better luck than the secret attackers.

3.3 Estimating time of thought

Estimating time of thought is crucial but difficult to do reliably. Some methods are discussed in §A.

The most important thing to get right is the ratio t/T. Note that when future security is an aim, then t can be quite large.

Cryptography standardization efforts, especially competition-style projects, like NIST's AES and PQC projects, have helped to boost T for the cryptography considered for standardization.

3.4 Diversified estimate for a

The **diversified estimate** a^* for attack probability a of strongest-link layering $C = C_1 \,\&\, C_2 \,\&\, \ldots \,\&\, C_n$ is

$$a^* = a_1 a_2 \ldots a_n, \tag{8}$$

which is the product of the attack probabilities a_i of the schemes C_i.

The diversified estimate applies if the schemes C_i have **attack independence**, meaning that secret attack events E_i are independent. (Recall events are **independent** if their probabilities multiply in the sense that

$$P\left(\bigcap_{j=1}^{s} E_{i_j}\right) = \prod_{j=1}^{s} P(E_{i_j}) \tag{9}$$

for distinct indices $i_1 < i_2 < \cdots < i_s$.)

See §B for some limitations to attack independence.

Given schemes $C_1, \ldots, C_n$ with attack independence, usage costs $U_1, \ldots, U_n$, with additive usage costs of $U = U_1 + \cdots + U_n$, attack probabilities $a_1, \ldots, a_n$, and damage D, then cost minimization is a discrete optimization problem: find the subset $M \subseteq \{1, \ldots, n\}$ that minimizes

$$\left(\sum_{i \in M} U_i\right) + \left(\prod_{i \in M} a_i\right) D. \tag{10}$$

If 2^n is small enough, then optimizing M is easy, given all the other inputs. The most difficult part of the analysis seems to be properly estimating a_i.

3.5 Compound estimate for a

The **compound estimate** a' for the attack probability a of $C = C_1 \,\&\, \ldots \,\&\, C_n$ is

$$a' = \widetilde{a_1} \ldots \widetilde{a_n}, \tag{11}$$

which is similar to the diversified estimate a^* for a, except that each factor a_i has been replaced by its thoughtover estimate $\widetilde{a_i}$.

The validity of the compound estimate depends on a further assumption, that the thoughtover estimates are independent, which in turn requires assuming that the t_i are independent.

3.6 Conversion to bits

For convenience, the previous variables are converted in Table 1 to a common unit of bits, defining five new variables **pain** p, **gain** g, **luck** l, **fame** f, and **hope** h. (This uses base two logarithms, $\lg(2^x) = x$, of probabilities, ratios, and other financial amounts, as needed. For example, each bit increase in gain halves the secret attack probability.)

Notation	Definition	Typical Range	Unit	Name
p	$\lg(D) - \lg(\$)$	[10,40]	Bits	Pain
g	$-\lg(a)$	[0,6]	Bits	Gain
l	$-\lg(-\lg(o))$	[-4,0]	Bits	Luck
f	$-\lg(t/T)$	[-5,5]	Bits	Fame
h	$l + f$	[-9,5]	Bits	Hope

Table 1: Converted-to-bits variables

The previous variables can be recovered from the bit variables by reversing the conversions, such as for expected loss like this:

$$L = \$2^{p-g}. \tag{12}$$

3.7 Estimating gain

Recall that gain g is $g = -\lg(a)$, where a is attack probability. Each type of estimate (thoughtover, diversified, or compound) for an attack probability leads to a corresponding estimate for a gain.

The bit variables tend to be additive. The diversified estimate a^* of a for $C = C_1 \,\&\, \ldots \,\&\, C_n$ leads to the **diversified estimate of gain**:

$$g^* = g_1 + \cdots + g_n, \tag{13}$$

where g_i is the gain for C_i. To estimate gain, we can use hope, which is luck plus fame. Similarly, a **compound estimate of gain** g' of is

$$g' = \widetilde{g_1} + \cdots + \widetilde{g_n}. \tag{14}$$

where $\tilde{g}_i$ is the **thoughtover estimate of gain**. As a function of hope h_i, the thoughtover estimate of gain can be computed as

$$\widetilde{g}_i = -\lg\left(1 - 2^{-2^{-h_i}}\right). \tag{15}$$

For $h_i > -\lg\lg(e) \approx -0.53$, the thoughtover gain can be approximated fairly well by:

$$\tilde{g}_i \approx h + \lg\lg(e) + 2^{-(h_i+1)}. \tag{16}$$

In other words, for high hopes, $h_i > 4$, the thoughtover estimate of gain is hope plus a constant plus a small number.

For $h_i < -2$, the thoughtover estimate of gain is well approximated by $\lg(e)2^{-2^{-h}}$. For low hopes like $h_i < -3$, the thoughtover estimate of gain g_i is less than $\frac{1}{100}$. Such gains might be so small that they are unlikely to cause the net cost to drop below the minimum threshold. Such gains might be so small that the usage cost can surpass the savings the gains provide to the expected loss (when used in a compound estimate).

3.8 Artificial numerical estimates

Some numerical estimates of the bit variables are provided in Table 2.

Scheme	Usage Cost	Fame	Hope	Gain	Attack probability
ECDH	2	2	0	1.000	0.50
McEliece	100	3	1	1.772	0.29
NTRU	3	1	-1	0.415	0.75
SIKE	10	0	-2	0.093	0.94

Table 2: Key encapsulation single-scheme estimates, with luck $l = -2$

These estimates are partly based on cautious hunches, with low fame estimates arising from large estimates for time of thought t by future attackers. These estimates are partly artificial, being adjusted to illustrate interesting non-trivial conclusions.

Experts in the specific schemes can improve these estimates by choosing better values of the input variables, based on their experience and evidence. Direct estimates of the natural variables t and T instead of the bit variable f would probably lead to more realistic assessments.

Table 3 evaluates the cost for each of the sixteen strongest-link layering of the four key encapsulation schemes. The minimal cost solution is layering

ECDH & McEliece & NTRU. In this example, adding SIKE to this slightly increased cost. The initial estimates in Table 2 for fame and usage costs were artificially tweaked to cause SIKE to be excluded from the optimum, in order to illustrate the possibility that the optimization of net cost can be non-trivial.

ECDH	McEliece	NTRU	SIKE	Net cost
-	-	-	-	1024
-	-	-	+	970
-	-	+	-	771
-	-	+	+	733
-	+	-	-	400
-	+	-	+	391
-	+	+	-	328
-	+	+	+	324
+	-	-	-	514
+	-	-	+	492
+	-	+	-	389
+	-	+	+	375
+	+	-	-	252
+	+	-	+	253
+	+	+	-	217
+	+	+	+	220

Table 3: Key encapsulation combinations, with pain $p = 10$

If the benefit was $B = 300$, then net benefit is positive as long as strongest-link layering includes both ECDH and McEliece.

As an alternative example, suppose that usage costs were lower, or damage were higher. In that case, including SIKE might lower the net cost (instead of raising it). Indeed with yet higher damages, even more layers of diverse cryptography (beyond the four in ECDH, McEliece, NTRU and SIKE) could lower cost even further.

4 Explaining the thoughtover estimate

This section describes a heuristic explanation of the thoughtover estimate.[7]

[7]This explanation revises previous work [Bro19] by the author of this report.

The explanation uses a simplistic model: specialized Poisson point process model, combined with general statistical inference.

4.1 Poisson model of cryptanalysis

Recall that independent public thought T is the total time spent trying to break a given scheme. Assume that

- the probability of breaking the scheme is a function π of T, and
- for two disjoint sets of independent thought with times T_1 and T_2, we have $\pi(T_1+T_2) = \pi(T_1)f(T_2)$. In other words, probabilities of breaking the scheme are independent for disjoint periods of thought.

These two assumptions imply the well-known Poisson point process model. There exists a constant A such that the probability of finding no practical attack in time T is:

$$P = e^{-AT}. \tag{17}$$

Call A the **attackability** of the cryptosystem. Attackability can range from 0 to ∞. If the attack does not exist, then $A = 0$. Otherwise, attackability quantifies how easy it is to break the scheme in a given T.

Well-known properties of the Poisson point process imply that $1/A$ is the expected (average) independent thought needed to discover an attack.

4.2 Inference by optimism

Suppose that no practical attack on the target cryptographic scheme has been observed after spending independent thought T trying to break the scheme. Assume that

$$P \geq o. \tag{18}$$

for some value o that we will call **optimism**. We call $o = 0.05$ **cautious optimism**.

A small o means that we recognize the possibility that the public attackers had the bad luck of not finding an attack. A too large o mean that we were overconfident of there being no attack.

(Statistical terms related to optimism are **confidence** and **significance**, but optimism seems more appropriate here.)

Substituting equation (17) for P in bound (18) bounds attackability A by

$$A \leq -\frac{\log o}{T}. \tag{19}$$

Putting $o = 0.05$ amounts to an estimate that the average time needed to find an attack would be at least $T/3.00$, after having tried and failed to find an attack in time T.

4.3 Independent secret thought

If a secret attacker has secret independent thought t, then the Poisson point process model says that the probability the secret attacker fails to find an attack is

$$q = e^{-At}. \tag{20}$$

In other words, q is the probability that the cryptosystem remains **secure** against the secret attacker with independent secret thought t.

Substituting the inference (19) into equation (20) bounds security probability q by

$$q \geq o^{t/T}. \tag{21}$$

The attackability A has vanished from this estimate.

The probability of a secret attack is $a = 1 - q$, which is bounded bounded by

$$a \leq \tilde{a} = 1 - o^{t/T}. \tag{22}$$

4.4 Thoughtover can over-estimate attacks

The thoughtover estimate is based on an upper bound estimate, meaning that the observed evidence is consistent with $a < \tilde{a}$. Nonetheless, as a prudent precaution, we consider it as an estimate for a, so $a \approx \tilde{a}$.

A newly proposed scheme C might actually be optimally secure, with $a = 2^{-128}$, but might have high thoughtover estimate of $\tilde{a} = 0.999$, because T is still small (C being so new), while t is much larger due to future attackers. In this case, the thoughtover estimate $\tilde{a} = 0.999$ is an overestimate for $a = 2^{-128}$. In other words, the thoughtover estimate of attack probability always starts high for new schemes.

A Methods to estimate time of thought

An estimate for the total time of public thought T is to sum the individual times of each person contributing to T. This summation assumes that each person has thoughts independent of other people, which is reasonable when considering undiscovered attacks.

The independent thought of a single person can be upper-bounded. The maximum number of years a single person can think about breaking a scheme, can be estimate by the age of the scheme C, and by the educational and work experience of the person. A typical person might have a maximum rate of thought per year of independently trying to break a given scheme C. An upper limit of 100 hours per year seems reasonable, accounting for the need to think about other things and also for exhaustion causing repeated thoughts that are no longer independent.

Also needed is an estimate of how many people have thought about breaking C, and the average amount of time they spend thinking about breaking C. Direct self-reports can be considered. Publication records might also help estimate times of independent thought. A partial attack on C, such as one that requires revising the scheme's parameters, can be regarded as strong evidence of thought.

Estimating secret thought t has extra complications. Secret attackers may not want even the size of t to leak: they may even try to deceive the public by implying t is too small or too large, perhaps to influence the public's decision to use the scheme C.

When aiming for future security, the secret thought t should include future thought. This future thought contribution to t could be quite large, and should be proportionate to the amount of time that future security is desired. Future thought is likely to increase with the increased deployment of the scheme C, but the most relevant estimations for the risk of secret attacks against C would assume that C is deployed.

Alternatively, one could estimate the ratio t/T directly, trying to compare a secret attackers capabilities against the public scrutiny. Such an estimate could be used as a check against the possibility that the estimate t and T are arrived at by different methods.

B Attack dependence

B.1 Clear overlaps between schemes

For some sets of scheme $\{C_1, \ldots, C_n\}$, such as key encapsulation, there might be clear overlaps. For example, ECDH, NTRU, McEliece and SIKE might all use the hash function SHA-2. Similarly, multiple signature schemes might all use the same hash function SHA-2.

Strictly speaking, such overlap rules out absolute attack independence. A single attack on the overlapping part, SHA-2 above, could break all the individual schemes.

To work around this, we can assume that overlapping parts are perfectly secure, making all estimates conditional upon the security of overlapping part.

B.2 Dynamic allocation of thought

A secret attacker targeting $C = C_1 \,\&\, \ldots \,\&\, C_n$ could also estimate the independent public thought $T_1, \ldots, T_n$, but could control $t_1, \ldots, t_n$ to optimize the success of finding a secret attack on C.

One possible allocation strategy is to choose t_i proportional to T_i. If the t_i are run in parallel (over the same time period), then the expectation is to break all C_i in the same average time period. This might minimize the switching resources between attack efforts. If the attacker adopts this strategy, then the attack probability is $(1 - o^{t/T})^n$, where $t = \sum t_i$ and $T = \sum T_i$. Surprisingly, the effectiveness of this attack does not depend on the individual T_i.

C Manipulating estimates

Qualitative recommendations and quantitative estimates are both vulnerable to intentional manipulation, or accidental bias. However, quantitative estimates are more open to review and correction.

D Diversity is needed to make agility work

The term **agility** means the ability to rapidly change the scheme in the event of a public attack. Diversity of schemes is needed in order to change to a scheme that is not vulnerable to the public attack. Diversity is needed to make agility work.

E Using estimation

The damage and the secret attack probability viewed together determine the most reasonable course of action in the given circumstances.

1. If a is optimally low, with $a < 2^{-128}$, and damage $D < \$2^{50}$, then the expected loss is negligible, with $L = \$2^{-78}$. In this case, there is no reason to improve cryptography, because any reduction in risk will be negligible.

2. If D is negligible, with $D < \$2^{-20}$, say, then $L \leq D$ because $a \leq 1$, which means L is also negligible. In this case, there is no reason to use cryptography at all, because the risk not using cryptography is negligible.

3. If D is high, say $D/\$ \in [2^{10}, 2^{40}]$, and a is non-negligible, say $a \in [2^{-30}, 1]$, then L could be non-negligible. In this case, there is reason to try to improve the cryptography, by lowering a to reduce the expected loss.

Note that the ideal $a = 0$ is impossible for most schemes, because a key-guessing attack has $a > 0$. A key-guessing attack can be regarded as secret in that the key-guesses are secret, meaning there is no simple patch for the public. The normal response to key-guessing attack is to allow them if a is negligibly small, such $a < 2^{-128}$.

Acknowledgments

A. Živković helped improve clarity and logic.

References

[BHK+19] Daniel J. Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. The SPHINCS+ signature framework. Cryptology ePrint Archive, Report 2019/1086, 2019. https://eprint.iacr.org/2019/1086.

[BLP08] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Attacking and defending the McEliece cryptosystem. Cryptology ePrint Archive, Report 2008/318, 2008. https://eprint.iacr.org/2008/318.

[Bro02] Daniel R. L. Brown. Generic groups, collision resistance, and ECDSA. Cryptology ePrint Archive, Report 2002/026, 2002. https://eprint.iacr.org/2002/026.

[Bro19] Daniel R. L. Brown. An optimist's Poisson model of cryptanalysis. Cryptology ePrint Archive, Report 2019/1465, 2019. https://eprint.iacr.org/2019/1465.

[CLN+19] Craig Costello, Patrick Longa, Michael Naehrig, Joost Renes, and Fernando Virdia. Improved classical cryptanalysis of SIKE in practice. Cryptology ePrint Archive, Report 2019/298, 2019. https://eprint.iacr.org/2019/298.

[DLL+17] Leo Ducas, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehle. CRYSTALS – Dilithium: Digital signatures from module lattices. Cryptology ePrint Archive, Report 2017/633, 2017. https://eprint.iacr.org/2017/633.

[HGHPW05] Nick Howgrave-Graham, Jeff Hoffstein, Jill Pipher, and William Whyte. On estimating the lattice security of NTRU. Cryptology ePrint Archive, Report 2005/104, 2005. https://eprint.iacr.org/2005/104.