Layering diverse cryptography to lower risks of future and secret attacks: post-quantum estimates

Daniel R. L. Brown[*]

July 28, 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 suggest that the security advantage of diverse layering can be worth the extra usage cost, thus advising a decision to layer diverse cryptography.

1 Introduction

A 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. A future attacker with a quantum computer would be able to break the ECC or RSA used today, by storing today's ciphertext and breaking them in the future. A current attacker who hides the existence of its Shor-running quantum computer can run a secret attack against ECC (or RSA) users. So, PQC hedges a set of possible future or secret attacks.

[*] `danibrown@blackberry.com`

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

1.1 Contributions of this report

This report suggests some methods to estimate the risk of future and secret attacks. The methods rely on strong heuristic assumptions, such as attack independence of diverse cryptography, and secret and future attackers have chances (over time) at discovering an attack equal to the current public attacker.

1.2 Limitations and caveats

The estimates have major limitations.

- The input variables are also estimates. So, the output estimates cannot be any more reliable than the input estimates.
- The input variables (observed data) are very few, resulting in a (necessarily) weak statistical inference, meaning a wide range of attack probabilities. On a precautionary basis, we take the highest attack probability as the estimate.
- Attacks are modeled as arising from a Poisson point process, each instant of thought having a probability (density) of leading to attack discovery.

Despite its limitations, the methods is simple, and provides a formalized upper bound estimates on risk. Arguably, the risk estimates are over-estimates. In other words, the risks might get exaggerated.

To repeat, because the statistical inference method is weak, and the input variables are not very reliable estimates, the overall method is statistically under-powered, and the risk estimates are counterintuitively high. Other estimation methods might take further considerations to reduce the risk estimates.

The estimates in this report sometimes lead to recommendation in favor of layering diverse cryptography. This benefit to layering diverse cryptography happens when trying to resist very powerful secret attackers, or more realistically, trying to attain very long-term security.

1.3 Examples of cryptography

For concreteness, this report focuses on some leading-edge types of key cryptography, mostly post-quantum. It considers four types of key encapsulation: ECDH, McEliece [BLP08], NTRU [HGHPW05], and SIKE [CLN+19]; and three types of signatures: ECDSA [Bro02], Dilithium [DLL+17], and SPHINCS+ [BHK+19]. The methods can of course be applied to other types of cryptography.

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,

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

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

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: $[\mathrm{ECDH}, \mathrm{NTRU}, \mathrm{McEliece}, \mathrm{SIKE}]$, or
- three signature schemes: $[\mathrm{ECDSA}, \mathrm{Dilithium}, \mathrm{SPHINCS+}]$,

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 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 available (with low cost, as explained later), and estimates the benefits of strongest-link layering.

2.3 Attack probability

Let E indicate the **event of a secret attack** against scheme C. Similarly, E_i is 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 $a = P(E_1 \cap \cdots \cap E_n)$.

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

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**[6] 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.[7] 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 $C = C_1 \& \dots \& C_n$ 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.

[4] Such as email or web browsing.

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

[6] Looser usage of the term risk as probability is not used in this report.

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

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 is needed, 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.)

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

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

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

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

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

Scheme	Usage Cost	Fame	Hope	Gain	Attack probability
ECDH	2	2.000	0	1.000	0.50
McEliece	100	3.000	1	1.772	0.29
NTRU	3	1.000	-1	0.415	0.75
SIKE	10	0.000	-2	0.093	0.94

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

These estimates are just examples. They 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.

The input bit variables have been rounded to the nearest bit. The output variables have been shown with greater precision, but this would be false precision under strict numerical analysis. A better treatment would provide input variables with higher precision than the output variable, or use ranges. Nonetheless, for the purposes of this example, the overly precise numbers illustrate the principles of the formulas can in some cases favor some combinations of layerings over others. For a more careful conclusion, precision issues should be considered.

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.

As noted earlier, because the input variables, such as luck and fame, were rounded to the nearest bit, the output variables, such as gain and attack probabilities, should, under proper numerical analysis, be treated with lower precision. In other words, with the numbers of above, the conclusions about

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 $p = 10$

inclusion or exclusion of SIKE in risk-minimizing combination are likely to be within the errors due to imprecision.

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.[8]

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

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

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 $\pi(T_1+T_2) = \pi(T_1)\pi(T_2)$. 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:

$$P = \pi(T) = e^{-AT}. \tag{17}$$

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 upper 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 estimated 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.

B.3 Independence and diversity are simplifications

The assumption of attack independence for diverse cryptographic is a simplification.

The pre-requisite condition for attack independence, the *diversity* a set of cryptographic schemes is treated as an boolean input variable to the estimation methods of this report.

In other words, whether a set of schemes is diverse is left by this report as a judgement call for the experts, depending on the tacit knowledge and experience of the readers. As usual, the quality of the output estimates is only as good as the quality of the input estimates.

Attack independence is a straightforward, but very strong, quantifiable consequence of the boolean estimate for diversity. Because attack independence is a very strong condition, a judgment concluding diversity is a significant conclusion, and should be made only with great care.

More precisely, the diversified estimate of attack probabilities accounts only for the event that no single attack affects multiple schemes (in the diversified set of schemes). In other words, the diversified estimates of attack probabilities ignore the event of a single attack affecting multiple schemes. Consequently, the real world attack probabilities should be higher than the diversified estimate, by approximately the probability of such multi-scheme attacks.

If the probability of multi-scheme attacks is high, then the diversified estimates are too low. In particular, the extent the diversified estimates support the security advantages of layering would be diminished. In other words, the arguments of this report offer no security advantage to layering non-diverse schemes.

B.4 Correlated attack probabilities

A very sophisticated model might try to infer correlations of attack probabilities on a variety of schemes.

The input data to the inference might use a heuristic measure of similarity, perhaps combined with a history of attacks on previous versions of the schemes.

This report does not attempt to quantify such correlations.

C Manipulating estimates

Qualitative recommendations and quantitative estimates are both vulnerable to intentional manipulation, or accidental bias.

Arguably, quantitative estimates are more open to review and correction. In other words, quantitative estimates are closer the ideal of evidence-based decision-making.

The estimates in this report are also rather open-ended, in that they depend on estimating diversity condition and on estimating the parameters t and T. It is possible to cheat on these inputs, of course. This open-endedness mean that the estimation methods might not be very well-suited for formally persuading third parties (as “propaganda” of the type discussed by Koblitz [Kob81]).

Instead, the estimation method might help inform a first-party or second-party decision of when to layer diverse cryptography. Even then, the estimation method should only be one factor influencing the decision. This report does not expect anybody to abandon all other forms of valid reasonings, including experience and intuition, but rather to consider these estimates to develop a more nuanced decision.

For example, the estimation methods in this report have largely re-shaped my own personal views on the probabilities of secret attacks. The high attack probabilities produced by this method are higher than what my past intuition has usually suggested. Even though I well understand the limitation of this report's estimation method (in particular, its being under-powered by limited input data), I cannot completely ignore the estimates either. In other words, I have re-visited and revised my own intuitions, questioning and doubting their bases.

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 from a newly broken scheme to a not-yet-broken scheme (not vulnerable to a public attack).

Diversity is needed to make agility work.

This report does not try to quantify the benefits of diversity to agility.

E More example ranges of risk

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

1. If $a < 2^{-128}$ (ideally low), and damage $D < \$2^{50}$, then the expected loss $L = aD$ is negligible too, with $L = \$2^{-78}$. In this case, there is no reason to improve the considered cryptography. Any further reduction in risk will be negligible (by comparison).

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

The ideal $a = 0$ is arguably impossible for a single user of nearly any scheme, because a single-user key-guessing attack has $a > 0$. A single-user key-guessing attack can be regarded as a secret attack because the attacker's key-guesses are secret. The single user has no mitigation, because the user does not know which keys that attacker will guess.

Fortunately, a single user can tolerate reasonably (but perhaps reluctantly) a value of a is negligibly small, such $a < 2^{-128}$.

Alternatively, a scheme designer may opt to to narrow the definition of attacks to those against a generic user, meaning attacks work equally well against any user (no matter their choice of key). In this latter generic user setting, it may be possible that $a = 0$.

Nonetheless, the thoughtover estimate always has $\tilde{a} > 0$, because its underlying model leaves no way to infer $a = 0$ with any statistical significance.

F Further informal discussions

The estimates in this report are simplistic: the formulas are simple, and the formal justification skip over many subtle details.

This section discusses, informally, some further issues beyond the simplistic model.

F.1 Simplified statistics

For a previous example of attack independence being used as a simplifying assumption in a formal probabilistic model, consider Bernstein [Ber20, §3.2]: "probability $p(M)$ of being publicly broken within M months. Assume for simplicity that these probabilities are independent across proposals".

Presumably, Bernstein assumes independence mainly to simplify the statistical inference of the function p from the observed data in the publication record. Presumably, Bernstein implicitly also means a carefully chosen set of proposals (so that nearly identical proposals are not to be regarded to have independent probabilities of getting broken). In other words, perhaps

Bernstein intends diversity to be consider before invoking the “probabilities are independent” assumption

This report attaches a formal condition of **diversity** as a **pre-requisite** to assume attack independence – rather than the simpler blanket assumption of [Ber20, §3.2] of independence across proposals.

The simpler blanket assumption of independence may be good enough for making inferences about the function p. The skew in the estimate for the function p might be tolerably small if the data (attacks on past proposal) has bias from non-diversity (undermining the assumption about independence).

By contrast, this report has a different statistical goal, estimating the security benefits of strongest-link layering for specific sets of schemes. For this goal, an explicit pre-requisite of diversity is critical.

To elaborate, suppose many nearly identical schemes are proposed at the same time. The saying that “great minds think alike” and the cryptographic competition format together make such simultaneous publication plausible. Being nearly identical, all these schemes might get broken by one attack. So, they would get broken in the same month M. When trying to infer the function p, a simplistic assumption of independence for all proposals would create a spike at $p(M)$. This spike in p would be artificial. Clearly, the spike could be corrected, by recognizing the attack dependence of the nearly identical schemes. In other words, hindsight recognition is applied. Non-diverse proposals skew the inference of the function p. The correction is fairly easy, if a heuristic, as an afterthought sophistication.

Generally, there is a limit to such model improvements by way of sophistication. It can result in over-fitting. A sophisticated model might assume too much, or allow too much. Assuming too much means the model may no longer be predictive, being based on false assumptions. Allowing too much means, that the confidence interval for inferences is too wide, so the model is under-powered.

F.2 Hints of non-diversity

Consider NTRU and McEliece. We can view NTRU as some variant of learning with errors, while McEliece is based on error correction. A common thread is *errors*. Consider the threat of a hypothetical single attack against this common thread. Such a attack might break both NTRU and McEliece. To the extent that this would be possible, then diversified estimates of attack probability is an under-estimate. In other words, this is an argument that

NTRU and McEliece should not part of a diverse set of schemes.

This potential threat suggests a smaller security advantage to layering NTRU and McEliece.

This report makes no estimates that quantify this threat. The quantified estimate for attack probabilities is based on an boolean input estimate: diverse or not. Perhaps more sophisticated methods can use quantified input estimates, estimates of how diverse a set of schemes.

Returning to threat mitigation (instead security estimation), this particular threat, that a given set, such as one including McEliece and NTRU, might not actually be diverse, has a simple mitigation: use a yet larger set, aiming for wider diversity, adding some schemes that seem to have no similarity to the suspiciously similar mainstream schemes (NTRU and McEliece, in this example).

Layering large sets of diverse schemes (as a mitigation to the suspected of non-diversity) might entail using some obscure schemes. In other words, scraping the bottom of the barrel. Even under the security estimates in this report, the return on investment for very obscure schemes, with low values for t/T, in a layered system can be very low, sometimes not enough to warrant the usage costs. The example with SIKE (which is not very obscure) at all.

In other words, the estimates this report sometimes favor layering diverse cryptography, they do not always favor layering diverse cryptography.

Nonetheless, a user very suspicious of the diversity of existing schemes, might, if only out of desperation, use a large number of layers, some very obscure. Such user places trust in the diversity and layering, but the basis would not be due to the estimates in this report, because very obscure schemes contribute negligible amounts to the security assurance.

F.3 Cumulative layering

Some cryptographic algorithms, such as block ciphers, apply multiple simple rounds. These rounds can be considered as layers. Often, the rounds have some diversity: some are linear, some are non-linear, for example.

Each round is weak on its own. Yet, with multiple rounds, the security accumulates. We can call this **cumulative layering**.

This report has viewed the strength of multiple rounds as the maximum of each round. In other words, it has considered **non-cumulative layering**.

If post-quantum cryptography schemes could be layered cumulatively, then layering would have even more security advantage. But there does not seem to be any evidence that cumulative layering is possible with post-quantum cryptography.

Indeed, being a form of public-key cryptography, it seems impossible to apply the cascading arguments for cumulative layering. In the case of public-key cryptography, an attacker can attack each public key separately, by finding its private key. Given all the private keys, any system of layering should be breakable.

For subtler types of attacks, such as distinguishing chosen-ciphertext attacks, perhaps cascaded layers might work as a form of cumulative layering. To repeat, there seems to no evidence of this, but one can hope.

F.4 Application to implementation faults

Thoughtover estimates might also apply to attacks on implementations.

Split the task of implementing cryptography into two parts: developing and reviewing. Developing means converting a cryptographic specification into an executable program or device that can generate known answer tests and inter-operate with other implementations. Developing the implementation is not free, so should still be included in the usage costs. Reviewing means spending time checking for security bugs and fault attacks against the implementation. Like other forms of cryptanalysis, it could be measured by thought, and separated into public and secret thought.

F.5 Revisions, evolution and tweaks of schemes

Consider a single cryptographic scheme that has an easily adjustable parameter, such as public key size. Suppose that, over time, one or more new public attacks against the scheme are published. With each new attack, the parameters scheme are revised. The revision is to increase the recommended parameters (such as large public key size), each time rendering the published attacks infeasible.

As example of such a history of adjusting key sizes, consider RSA public-key cryptography, which has gone from 1024-bit public keys to 2048-bit or even 3076-bit keys.

The thoughtover estimates do not attempt to directly account for such a history of adjusting parameters. Rather, it is based on whether scheme with a fixed set of parameter will be subject to attacks or not.

Nonetheless, a history of revised key sizes, could be taken a strong evidence that thought has been applied to the latest recommended key sizes. For example, in the case of RSA public-key cryptography, efforts to factor 512-bit, 768-bit and 1024-bit numbers is evidence that thought has been put into trying (and failing) to factor 3072-bit numbers. In other words, a history of revisions contributes to the input variable T,

That said, two rather opposite interpretations of the history might be taken for contribution to T.

- As outline above, a public attack on a previous version of a scheme can be interpreted as a failure to find a attack on the latest version of the scheme. The public attack is therefore evidence for some contribution to T.

- By contrast, public attacks on a previous version of a scheme might indicate some kind of momentum of cryptanalysis that could potentially continue, as discoveries of weaker version of a stronger attack. With this view, one might wish to reset the clock, and rate a very value for T for the recently fixed version, as if it were an entirely new scheme.

Clearly, any new, innovative changes to a cryptographic scheme should be considered as quite new, and should not inherit any contributions of T to older versions.

F.6 Eureka moment of inspiration, or evolution?

The thoughtover estimate is heuristically based on a quantitative assumption the Poisson point process. Qualitatively, this could be characterize attack discovery as Eureka moment of discovery. Time is needed to think, but the successful thought occurs in an instant, a moment.

It might be more realistic to model attack discovery occurring not in a moment of thought, but gradually over some time. It might begin with a hunch: a researcher begins thinking about a system and first gets a vague feeling that there must be an attack. Just before the attack is discovered, the researcher sees the light at the end of the tunnel. The researcher might also make several failed attempts at attacks, before finally succeeding. Then the researcher writes up the attack, just to be completely the idea is correct.

This process may be more realistic, but it may also be hard to quantify, especially over many researchers. It also introduces more parameters into the model, making inference even more difficult. Instead of attackability, there might also be hunch-ability, failed-attackability, and so on.

F.7 Under-powered over-estimates

A major reason that thoughtover estimate of attack probability is an over-estimate is that it is based on an under-powered statistical model.

This under-powered model looks only at two aggregate input numbers t and T, which contribute to ratio t/T. The numbers t and T might be based on reasonable, ample evidence, but nonetheless all this evidence is condensed into two numbers, which arguably amounts to a statistical bottleneck.

The Poisson point process model also uses only a few assumptions. Using more assumptions, as long as they are realistic, might narrow the model, allowing a narrower and more precise inference.

This under-powered model underlying the thoughtover estimate causes it to infer wide ranges for the attack probabilities. As precaution for users, this report takes the highest attack probabilities in the wide inference range. The highest attack probabilities are potentially over-estimates.

F.8 Unquantifiability of thought?

A natural objection to thoughtover estimates is that thought should not, and perhaps cannot, be objectively quantified, because it is too amorphous, and perhaps too sacred.

Yet, how else might an attacker find an attack, other than by thought?

Some common ideas also suggest that is not such a stretch to attempt to quantify thought in units of time (across multiple people).

- Easy problems are described as needing only a moment of thought.
- Difficult problems are described as defying the long and hard thoughts (of experts).
- Sometimes, two heads are better than one.
- Mathematics students practice problem-solving in time-limited examinations.

F.9 Philosophy of thought

Descartes argued that, when faced with great uncertainty about existence of things, thought is the one thing we can be most certain about. To address the very uncertain risk of future or secret attacks, it might make sense to think like Descartes, and turn to thought as a source of certainty.

The philosophy of materialism, by contrast, does not consider thought as so fundamental to existence, but rather a second by-product of existence. A materialist might see the model used for the thoughtover estimate as too indirect, hinging on a secondary property of the material, and might prefer a model such as [Ber20].

F.10 Psychology of thought?

Behaviorism is an approach to psychology that emphasizes observable and measurable behavior, instead of non-measurable internal like thought.

The thoughtover estimate certainly does not adhere to behaviorism. A behaviorist might prefer an approach along the lines of [Ber20], assessing the what cryptographers and cryptanalysts typically actually achieve, regardless of what internal thought process they use.

F.11 Thought experiments

Physics uses the term **thought experiment** (as a slight pejorative) to describe an experiment carried out entirely in thought. The legitimacy of thought experiment is at best the match between thoughts and reality, which is arguably based on tacit knowledge such as experience.

Perhaps, the model used for thoughtover estimate can be considered a thought experiment. It asks the user to think about attackers (the times of thought t and T), and then draw a conclusion.

F.12 Variable attackability

The thoughtover estimate is based on a notion of attackability, the ease of a finding an attack. Furthermore, attackability is considered individually for each cryptographic scheme.

Attackability is not directly observable, even upon attack discovery. It is a subjective, psychological quantity measuring how easy it is to find an attack that exists.

The point of of this variable attackability is that some problems may require more thought than others.

As an example, consider problems in a mathematics textbook or student examination. In some case, some problems are rated as more difficult than others. Sometimes, the difficulty is proportional to the length of the known solution, but not always. Some easy problems have long solutions because they are straightforward but tedious. This variable difficulty of mathematical problems (with known solutions) is well-established, and used quite commonly.

F.13 Finite total thought

As a further assumption, one might somehow estimate that, for a given cryptographic scheme, that there is only a finite total amount F of thought possible in trying to find an attack. By definition, $t + T \leq F$, so $t \leq F - T$.

The main parts of this report do not use such F. The only limitations on the effectively thought towards attacking a scheme is the attackability A.

Nonetheless, sand reckoner arguments support that such an F ought to exist. Estimating a number for F, with the same precision and confidence as one estimates t and T is an altogether more ambitious aim.

Nonetheless, one might intuitively infer F from two observations of oneself: firstly that one can run out of one's own attack ideas, and secondly that others tend to have similar ideas to one's own ideas. From these tow observations, one might extrapolate these limitations to the population at large and infer a value for F.

The distinction between T and F is that T represents actual total public thought-expenditure, whereas F represents total thought-exhaustible-potential, or capacity.

An estimate for F very close to T implies small upper bounds for t. This then makes the thoughtover estimates for attack probabilities much smaller.

To be clear, if one estimates $T \approx F$, one must clearly assume that that the total populations possible thoughts towards attack the cryptography scheme have been exhausted, not everybody has spent all their available time think how to attack the cryptographic scheme (which is clearly not the case).

Estimating that $T = F$ has the same practical impact as inferring that $A = 0$.

The philosophical difference between $T = F$ and $A = 0$ is large. Some super-intelligent beings might surpass humanity, and not be bound by F. Against a super-intelligent adversary, we should raise the value of F, which means the upper bound $t < F - T$ gets much higher, as do the thoughtover estimates for attack probabilities.

So, an inference $A = 0$ is much stronger than an estimate that $T = F$. Inferring that $A = 0$ amounts to saying that not even super-intelligent beings will be able to find an attack: $A = 0$ means that no attack exists.

Even if no attacks exist, how can we infer this from t and T. How can we infer a limitation on super-intelligent beings based on the efforts of beings of normal intelligence?

The best way to address of ruling out super-intelligent beings outwitting us is the tool of mathematical proofs. Without a proof, a claim is at best a conjecture, and should be inferred as a nonzero probability of being wrong. This model of this report accommodates this nonzero probability by always inferring that $A > 0$.

F.14 Rate of thought (public cryptanalysis)

Consider a model with a unknown parameter, a function τ that measures the **rate of thought** per calendar time. The function τ, being unknown, is something we try to infer from the data.

By rate of thought, we mean merely that public thought is:

$$T = \int_x^y \tau(m)dm \tag{23}$$

where m is time, x is the when the first public attackers began thinking about the cryptographic scheme (which might be before the scheme was published), and y is the time now (when we are trying to estimate attack probabilities). In other words, $y - x$ measures the age of the cryptosystem.

The rate of thought τ can vary with time. For example, $\tau(x)$ might be initially high, as the initial public attacks might be the inventors of the scheme. Then $\tau(m)$ might drop, as the inventors run out of attack ideas. After scheme publication, τ might increase a little, but then drop, with the scheme becoming somewhat forgotten after initial publication. Then, perhaps, some standardization or cryptography competition might revive interest in the scheme, causing τ to increase to a much higher level. If the cryptosystem gets deployed widely, then τ might increase to an even higher value as many more public attackers will have incentive to think about attacks.

For simplicity, one might also assume that τ is a constant function, and also assume the constant is same for all cryptosystems. This forgoes any attempt to account for varying review of cryptosystem, to distinguish obscure from popular. But it has the simplification that public thought T can be estimated easily using calendar time.

F.15 Survival functions

Let $S_A(T) = e^{-AT}$. This is the probability of no public attack after public thought T. Measuring T in time (but not calendar time), the function S_A can be considered a **survival function** [Wik21], because once an attack is published, the cryptosystem's life is over. So, $S_A(T)$ measures the probability of surviving to time T.

This report assumes that the survival function is a Poisson distribution. To be repeat, and to be fully clear, this is an assumption. Perhaps some other survival function, say $S'(T)$ is more realistic.

F.16 Effective effort

There is a sense (detailed below) in which this report defines "thought" T such that it can fit any existing survival function S' by a change of variables.

To distinguish this tautological meaning of T from the psychological meaning of T, we can introduce another quantity **effective effort**.

For simplicity, suppose that the survival function $S' : [0, \infty] \to [0, 1]$ is bijective, and fix some value $A > 0$. Define **effective effort** as $E = S_A^{-1}(S'(T))$. This implies that $S'(T) = S_A(E)$.

The survival function is a Poisson distribution as function of effective effort E.

(If S' is not bijective, then the range of possible E and T can differ, giving effective effort E a finite range $[0, F]$, where $F = S_A^{-1}(S'(\infty))$.)

The notion effective effort is a tautology, because effective effort amounts to re-parameterizing the survival function such that it becomes a Poisson distribution.

This report's model can be seen as an assumption that thought, something that all intelligent beings possess and that this report tries to quantify as numbers t and T, is equal to effective effort.

A more sophisticated model might try to distinguish between thought and effective effort. It might to do, so for example, by trying to infer a single survival function from many different schemes (similar to Bernstein's suggestions in[Ber20, §3.2]).

F.17 Subverted cryptography

Consider a **subversion attacker**, described as follows.

First, the subversion attacker first tries to secretly break a cryptographic scheme C'. Suppose that subversion attacker finds a secret attack on C', after spending secret thought t. The fact that the subversion attacker had to spend thought t to find the attack is related to the attackability A of C'.

Next, the subversion attack proposes the scheme C' to the public for usage and standardization. In other words, the subversion attacker tries to get the public key to use a scheme C', the subversion attacker can break.

To address the threat of subversion attackers in the thoughtover estimates, estimate a value of t that accounts for the possible time a subversion attacker could have spent to trying to break C' before C' was published. In other words, backdate the age of C'.

Fortunately, the subversion attacker's thought t already spent to discover the secret attack no longer increases with time. By comparison, T increases. As t/T gets smaller, the secret attack gets more likely to be discovered by a public attacker.

F.18 Ramp-up time

Independent thought is not meant to count the time and thought that each potential attacker spends learning about a cryptographic scheme.

This time, sometimes called **ramp-up** time, is not really independent, because the attackers are learning about the cryptographic proposal from somebody else's description.

Ramp-up time has a real world cost. The cost is the number of schemes times the number of attackers (times the non-memorability of each scheme. This report does not include rump-up time in its costs.

Very complicated and intricate cryptographic schemes could have very high ramp-up time. A plethora of schemes with high ramp-up might swamp attackers. However, if they have low attackability, once one attacker gets over the hill of ramp-up, finding an attack would be easy.

High ramp-up times, say for a scheme like SIKE, can be accounted for thoughtover estimate by imposing a penalty in the estimates for public thought T. Fewer researchers have enough time get through the ramp-up time, so T can be estimated as smaller.

F.19 Lapsed thought from attacks

Each time a cryptographic scheme gets broken by a public attack, the public thought T spent on it no longer contributes to thoughtover estimates on unbroken schemes.

In this case, we label the time T spend on the broken scheme as **lapsed** thought.

Lapsed public thought has value, since it has saved the public from using an attackable scheme any longer. In hindsight, it would have been better if the lapsed thought T had instead been applied to some other scheme that has not yet been broken.

A potential mischievous denial of service attack is possible. This attacker might propose many insecure cryptographic schemes. Each might consume some of the limited supply of public thought. Eventually, each weak scheme will get publicly broken, but the thought spent will be lapsed.

This attack is trying to use up the available public thought for the remaining unbroken schemes. This weakens the thoughtover estimates, and more dangerously, might be able hide secret attacks, because the public did not have enough to time to discover them.

Perhaps the mitigation to this denial of service is that each proposer or proponent of a cryptography scheme C has an extra duty to spend thought trying to break C, to avoid swamping the general public attackers.

F.20 Conditional probabilities

Bernstein [Ber20, §3.2] defines a probability $p(M)$ of a scheme "being publicly broken within M months" (of its publication, presumably). The idea is that p is an increasing function of time, presumably (due to the word "within"). Let $p(\infty) = \lim_{t\to\infty} p(t)$.

Bernstein uses a conditional probability

$$\frac{p(\infty) - p(36)}{1 - p(36)} \tag{24}$$

to calculate the probability that a cryptographic proposal is "breakable", after having seen that the cryptographic proposal was not publicly broken in 36 months. A less ambitious calculation can be used for the probability that a scheme will not be broken, within, say 2400 months, by a similar formula:

$$\frac{p(2400) - p(36)}{1 - p(36)} \tag{25}$$

The function $S(t) = 1 - p(t)$ seems to be a survival function, where t measures calendar time.

Perhaps the survival function is a Poisson distribution. In the notion of this report, the survival function would be a Poisson distribution with respect to calendar time, if the rate of thought τ is a constant function.

In this case, we would have $p(M) = 1 - S(M) = 1 - e^{-AM}$, and the conditional probability in (25) works out to

$$\frac{p(2400) - p(36)}{1 - p(36)} = \frac{1 - e^{-2400A} - 1 + e^{-36A}}{1 - 1 + e^{-36A}} = 1 - e^{-(2400-36)A}. \tag{26}$$

So, in a Poisson distribution, the probability of an attack in a given time interval simplifies to a value depending only the length of the time interval. For an arbitrary survival function, a full conditional probability calculation is necessary.

F.21 Prescient attackers

A **prescient** attacker is an attacker that finds an attack that public attackers would need an infinite amount of time, or an infinite amount thought to find. Bernstein's value $p(\infty)$ from [Ber20] can be interpreted as the probability of a prescient attacker. Bernstein also labels a scheme with a prescient attacker as "breakable".

In particular, a conditional probability such as (24) always results in 1 if $p(\infty) = 1$. Therefore, the conditional probabilities are only meaningful if $p(\infty) < 1$. If $p(\infty) = 1$, then the conditional probabilities are 1, and therefore vacuous.

To model a prescient attacker in the thoughtover estimates, one can try to put $t = \infty$ into the estimate. This gives $\tilde{a} = 1$.

The intent of this report is to disallow $t = \infty$, as infeasible and impractical, per normal conventions and notations of finite numbers. The principle is that a secret or future attacker does not have the capacity for an infinite time of thought.

It can be argued [9], that despite these conventions, this report can be read as saying, by its cautious inference of $A > 0$, that the probability of a prescient existing is 1. This is similar to putting $p(\infty) = 1$ in [Ber20].

What counts as prescient attacker also depends on what counts as an attack, or what counts as broken. Consider three examples.

- Any cryptographic scheme with a finite key space. A prescient attacker with infinitely much time can search the whole key space, and break each instance of the scheme. Perhaps, this should not be considered as attack, in that, that attacker does not end up with an algorithm that can break any instance.

- A preprocessing attack [CGK17] on the discrete logarithm problem (DLP) is possible, doing an initial work of $n^{2/3}$ to be buy an ability to compute discrete logarithms at cost of $n^{1/3}$. Should this count as a prescient attacker? Should DLP schemes be downgraded from $n^{1/2}$ security to $n^{1/3}$ security when we want resist a prescient attacker?

- Keyless hash functions have collisions. Collisions imply the existence of efficient algorithms. Should a prescient attacker be considered to have these algorithms? A long line of research refuses to consider keyless hash functions, considering their security to be undefinable. Yet another long line of research considers such hash functions secure (and even models keyless hash functions as random oracles!). So, this is an old and controversial issue. Rogaway [Rog06] has gone a long way to reconcile these opposing views.

This report takes the view that existence of prescient attackers is tolerable, because it makes no practical difference to real world security.

Mathematically, it is sometimes possible to prove that all attacks, including prescient attacks, will fail. Consider Shannon's' security proof of the one-time pad. Such proofs are not based merely on the failure to find attacks in a finite amount of time, but something more substantial.

[9] as in, https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/OpFVbuMYk8c/m/yF6p-TwSAwAJ

F.22 Random cryptographer model

In the random cryptographer model, we assume that there is a single random variable for the attackability A. The idea is that there exists an unknown distribution of competencies of the cryptographers who propose cryptographic systems. Let b the probability density function for random variable A.

Any cryptographic system corresponds one sample of random variable A. We then assume a separate survival function $S_A(t) = e^{-At}$ for each cryptographic scheme. For an average scheme, we average over the distribution of A, to get a single survival function for all schemes:

$$S(t) = \int_0^\infty b(A) e^{-At} dA. \tag{27}$$

Bernstein's theorem of monotone functions [Wik20] implies that a given survival function S has this form if and only if it is a **total monotone** function, meaning that $(-1)^n S^{(n)}(t) > 0$ for all $t > 0$.

Consequently, given function S it may be possible to determine its consistency with the random cryptographer model.

F.23 Inferring survival function

There are known statistical methods to infer survival functions from given lifetimes.

One method is the Kaplan–Meier estimator. This estimator seems to have the peculiar property that the property that conditional probability of survival past the age of oldest observed death is one. Applied to cryptography, if McEliece is the oldest unbroken cryptographic scheme, then the Kaplan–Meier estimator would predict that it is unbreakable.

The Kaplan–Meier estimator for the survival function is not smooth. Perhaps one can instead for a smooth survival function, perhaps total monotone function, that somehow fits well with the Kaplan–Meier estimator. From there, one can extrapolate the tail of the survival function, and so on. That might be quite interesting, although it might be unclear what the statistical confidence would be the tail of survival function.

F.24 Falsifiability

Ideally, a prediction, such as a probability estimate, should be falsifiable.

But falsifiability of probability requires an ability replicate the probability experiment. This report considers each cryptographic scheme as unique, making replication difficult, and therefore falsifiability difficult.

Moreover, this report also tackles the issues of secret attacks, and attacks far in the future. This issues also make falsifiability difficult, at least in the short term, based on public information. Furthermore, this report relies on input data expressed in time spent thinking, whose estimates are also quite difficult to falsify.

Making non-falsifiable predictions is generally problematic, because there is no easy way to falsify faulty predictions. Nonetheless, this report's main recourse is to use some reasonable models, such as Poisson point process.

Some replication is possible in the random cryptographer model, if we fix rate of thought for all schemes. Then we can follow Bernstein's strategy [Ber20] to collect data about attack probabilities over time, to estimate a curve for the function $p(M)$. If this curve is not totally monotone, then the observations may be able to falsify the model in this report (or a fixed rate of thought functions for all the functions).

F.25 Tapping tacit knowledge

Tacit knowledge refers to the accumulated experiences and wisdom that are not easily expressible. To a limited extent, time of thought is meant to estimate this.

A fairly common quantified way to tap tacit knowledge on an issue is to do a survey of experts. Identify some experts, who have worked on a particular open issue (whose future is difficult to otherwise predict). Then ask these experts about their predictions on this issue. Take an average. This method has been used for the estimate the risk of a quantum computer breaking ECC by 2030 [MP21].

When extending beyond acclaimed experts, this method is sometimes called crowd-sourcing, and in economics, a rational market. The advantage is that the surveyed people may somehow have knowledge that supports an educated guess on the issue. This extension has the advantage of eliminating the bias from the potentially biased selection of "experts", but has the disadvantage that many members in the crowd may be already be following the crowd. This feedback could result in a kind of group-think, and may be subject to various other biases.

F.26 Thought as unbounded recursion?

Consider an attacker who tries to be very thorough, by enumerating all possible algorithms, and testing each to see if it results in an attack.

This hypothetical attacker can be expected to have great difficulty, because most algorithms will be nowhere close to being an attack algorithm. Most of this hypothetical attacker's efforts will be wasted.

Nonetheless, real world attackers will occasionally think up possible attack algorithms. Not being sure if the attack algorithms will work, the real world attackers might need test the algorithms by implementing them on a machine.

Let us model the attackers thinking up such attack algorithms, as an innate and intuitive ability to scan through many attack algorithms very rapidly, somehow honing in on those algorithms more likely to result in attacks.

In this case the thought T can be considered as an implicit measure of the public attacker's effort to scan through possible algorithms for those that could be attacks.

In terms of computability theory, each possible attack algorithm can be regarded as a primitive recursive algorithm, whose runtime can be bounded in the advance. A search through all possible algorithms, however should be considered as unbounded recursion. In other words, it is recursion by minimization, finding the first algorithm that breaks the target cryptosystem.

Loosely speaking, the thought T somehow measures the progress through this step of unbounded recursion. Other resources, such as machine runtime, represent bounded recursion (which is more predictable).

Acknowledgments

A. Živković helped improve clarity and logic.

The second version of this report includes revisions, and an entirely new section §F (which began as a response to criticism from D. Bernstein).

References

[Ber20] Daniel J. Bernstein. Cryptographic competitions. Cryptology ePrint Archive, Report 2020/1608, 2020.

https://eprint.iacr.org/2020/1608. F.1, F.9, F.10, F.16, F.20, F.21, F.24

[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. 1.3

[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. 1.3

[Bro02] Daniel R. L. Brown. Generic groups, collision resistance, and ECDSA. Cryptology ePrint Archive, Report 2002/026, 2002. https://eprint.iacr.org/2002/026. 1.3

[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. 8

[CGK17] Henry Corrigan-Gibbs and Dmitry Kogan. The discrete-logarithm problem with preprocessing. Cryptology ePrint Archive, Report 2017/1113, 2017. https://eprint.iacr.org/2017/1113. F.21

[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. 1.3

[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. 1.3

[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. 1.3

[Kob81] Neal Koblitz. Mathematics as propaganda. In L. A. Steen, editor, *Mathematics tomorrow*, pages 111–120, New York, 1981. Springer. C

[MP21] Michele Mosca and Marco Piani. Quantum threat timeline report 2020. Technical report, Global Risk Institute, 2021. https://globalriskinstitute.org/publications/quantum-threat-timeline-r F.25

[Rog06] Phillip Rogaway. Formalizing human ignorance: Collision-resistant hashing without the keys. Cryptology ePrint Archive, Report 2006/281, 2006. https://eprint.iacr.org/2006/281. F.21

[Wik20] Wikipedia contributors. Bernstein's theorem on monotone functions — Wikipedia, the free encyclopedia. https://wikipedia.org/wiki/Bernstein's_theorem_on_monotone_functions, 2020. [Online; accessed 11-June-2021]. F.22

[Wik21] Wikipedia contributors. Survival function — Wikipedia, the free encyclopedia. https://wikipedia.org/wiki/Survival_function, 2021. [Online; accessed 31-May-2021]. F.15