Wedel, November 25, 2015

Achieving Post Quantum Computing Security and negligible Adversary Advantage in Hybrid Ciphers using a "NADA-Cap"

Michael Anders
University of Applied Sciences Wedel,
Feldstraße 143,
D 22880 Wedel
FRG
e-mail: Michael.Anders@academic-signature.org

1. Brief description of the concept NADA-Cap

NADA stands for No Access Data Available. A NADA-Cap conceals all information in a hybrid cipher file and renders the cipher indistinguishable from noise in toto. The adversary cannot extract any information from the cipher file except its file size, which necessarily gives an upper bound for contained entropy.

In fact the adversary can't even conclude that the file is a cipher text at all. NADA! Only if the adversary knows the NADA-Cap key, he/she can expose formatting and cipher info, which are traditionally given in the clear and learn about the nature of the file as an enciphered document.

2. Relation with the cryptographic concept "advantage"

In cryptography there is the theoretical concept of the "advantage of the adversary". This is a fairly intricate concept and is usually applied to nude cipher texts. I will explain and use it here in a very simplified form.

Assume you produce a cipher file f1 of a plaintext known to the adversary and a true random file f2 of roughly the same length. Given specific limits on execution time and storage space, which exclude exhaustive brute forcing, the adversary's advantage is zero, if he cannot distinguish whether f1 or f2 is the cipher. After n unsuccessful guesses in a k-bit keyspace the advantage necessarily rises to $n/2^k$, which can still be called negligible. If, within the given limitation, the adversary can definitely tell which of the two is the cipher, the advantage is one. The advantage is somewhere between 0 and 1 when the adversary(a probabilistic algorithm) can tell them apart with a certain corresponding probability. A specific symmetric cipher is considered to be safe, if the adversaries advantage is only negligibly different from zero.

In this paper, I will try to apply the concept in a simplified form to real world public key hybrid cipher files, which, compared to nude ciphertext, contain additional information usually given in the clear. A standard hybrid cipher can thus easily be distinguished from noise. Even when the underlying symmetric cipher is safe and thus in itself has negligible adversary advantage, the whole public key hybrid cipher will usually be of adversary advantage negligibly different from one. Thus they would, as a whole, be qualified as insecure ciphers according to a strict application of the rules.

A well accepted narrative, however, is that practical hybrid cipher files cannot conceal their nature and thus the strict rules would not apply to them.

3. Two practical ciphers

If you google for "open source public key cryptography", you'll find links to Academic Signature and GnuPG. Let's have a look at the cipher files of these tools respectively.

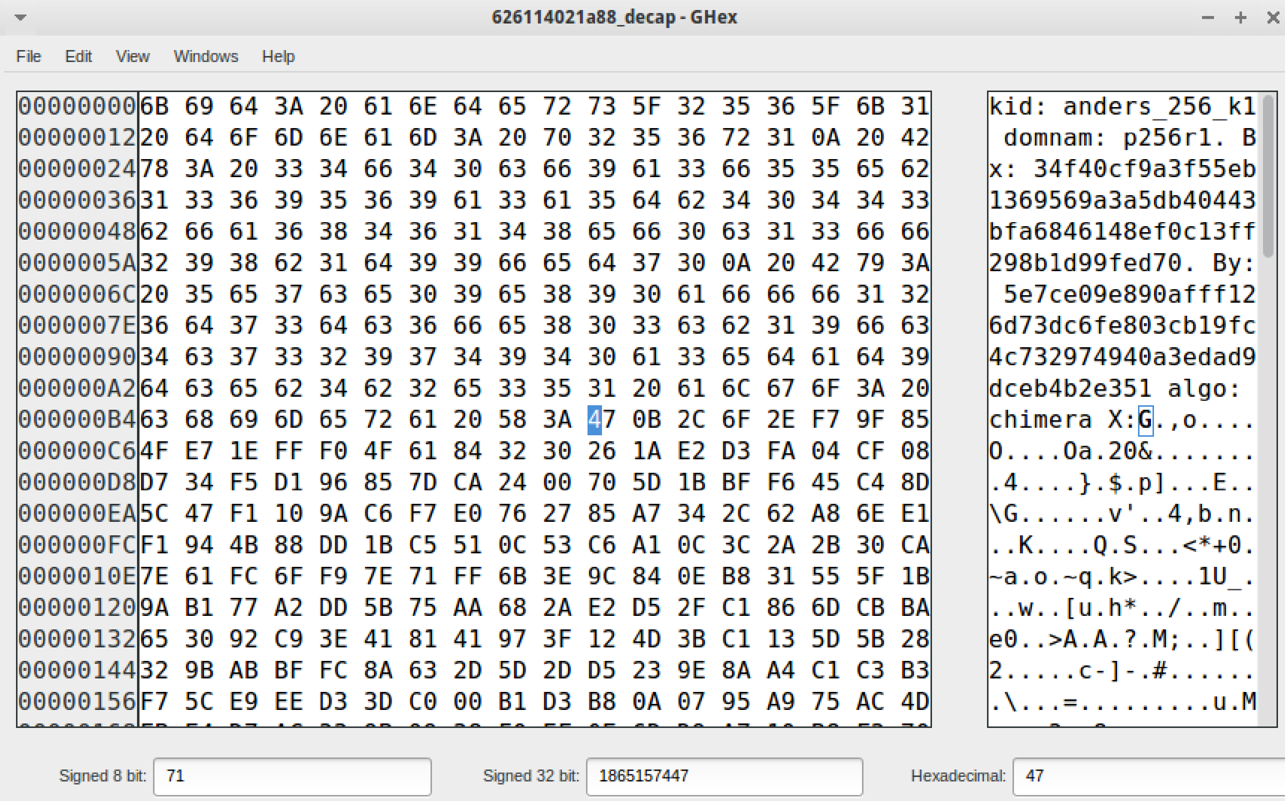
In GnuPG the standard cipher of a small precursor of this text looks like this:

```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v2

hQQKA9zqG3xrE27PEA/sC/Nj0Iigexp6Pg4oCNkCBck32+zZE9gbP5t5BOCGIKEo
sQqwdvlUSNbNjm87KYti0SsOb4spB51aywtxwwWdcAwTRBConI9FJ4xNy2dS6Pms
c8rZlOmT07PpjmH6hUDGlSTr/bAatH198iGgwxJ20tgowcRZEE6Sj0MihYsQEOZ3
reYbSZ280bYGXm6tEh0D4q83D3S3xEvh8n6TYMdZ91J+yB5qBdKZi5qoAebtWifK
EFl4haV5Z9pSJw/3l893D5IXey0hZSTSpSafGsZrgMUU2mBACtkzaQc7hMDYb/4Z
dagFFqREbLI7al4y099unCImSeEPi18ktWByNWDptpo9W6M5nRTeEKyXyGqKruVT
7nGPAuOPHaPtVX0Jlrq8K59U4qyrPgRXx5DQDwYSm+lk+PjVcZvk6CADh1eUxNgd
vaIvVZTAq/YqGaNlHjEEWUZRKqJgR31gG2ahr38CdEK9QsU9QXQyuALENtP/woeq
0dgZy8iPS0MvuBVWKmU3N7CukXhBqjiRNESZXxNKKCTpQ8u1szdv0d2QLwCuIiwR
/1h/itDH2i7Udn/Y6F+SLwgTvbdJVNUYVSw2sBo1WqCHkJABrzQMBAcft6OKCBwG
DbMctnTQ8EMeImGYFC0meWw95o7kYf93N35GjtpY4KijzMt22bw/XLrSE51KD+0U
01DoCkoo+7KLGtjO9kP87xi7uhodBeE0+jLDOGFhegojR6uzk6fA6mp7CxByM3y2
/rj58rnbEwLwv2z/kvp+49AH+RZaGYA1dEaVNpHpDskFAnISaIiM/7hg3tOFAGtW
XerCqhXFmW4afmXVPMQAs+QjHkINwUhg1tqdQc3IL1jZeosTq/xJbcJYXDTf424Z
adVntkxu9hofCEGQqW0WsJurjXp/kHXxs9xmcfPI009MpHUu2T42zsQ5SBKMCaok
nI1dZ3stl75n4DKWAD6xkxdxBQZuXY3536VdKOZ55zrb73TSpIyJO6za36kmh5Gp
Ia460hloB/8BQRY22S2tnoMMX87Lz3SaYpf4ZyswRt5f7/sEXNX+sMx3EyNSCkNL
m4DnlbJfTWLo7NkZyiJ41EK9wTAeqPq78+KTunYxOCb6NplMFZOTl9teq8wMnhHT
E/JKi7liAleMVr3h3eO3dzyX8rwppbbbNf3/vkjnE+fZytXfMd+4qfTpef6HgO1+
cgGCnWvN90jNV5CzildfXi7q6BlJQT32rIjRV6caFYprYkYTTUAbOzrOeulnrn5w
xSWiq4lz5DYBG/FxNGYSJCwnjhoQXZiJhvmM2plC15VVlBJN1vLjE1VkKO2YT90g
DaSHZ0KSGidDrYjf1sNigg+E29JaEvxoAqks4i7SwOIBhf4kP5C0UszzSag4plvP
9IFfX4ENnTPikCboknqU0T0ctn4MlcaAXgP2GvT7FZOT3fQV7xnnZ67YCZK5/aKB
GbCKMS5CC+a7+CgVgsMcvx+Vsv8PJLgECl0QCS4+phu8WwGd54gSKrlBOd+l299j
rGfVi+QJF+3TNVfAbvsw5v4OI3Sn9kt9BZzGeBnVTZKp3rx3JQKJJpStSJ38OOWL
u8ej85VdqUX+qx1CF0RvsLDO5sPa5SJdyHonnlRNzgTd1jpcOrJDIIZsfi673GLF
9TtRLCM0QTpCBeiZXLayb2rGBHZV0MQu+ZvtbZS62PuZ30y6UfFAJ7KQ4+76nmuy
6r04ZD7qEZK3nua3+/DbetAZqHQqfx7TNSizRhMwIf7zGBKLjYw/WtP12Ctnefhk
Q+Nsg5GbNiewzuIeEEqQCY0qLZT3PGV2/voUsi8KsltLPbzu9Y+szK628ZsBNSN3
P17w6/rqwPIxE8iFBkIfoQUKuh09UsxA7gpdU5QBcUPTpkvqzXOr24MQpSMmTqJw
yvB17GKeFEr1jD7gAVaWqgJs
=w4QO
-----END PGP MESSAGE-----
```

The file clearly identifies itself as a GnuPG v2 cipher. After this identifying header there is content in radix 64 encoding, part of which is not cryptographically protected and contains access information in the clear. You can find the format specification in RFC 4880. Aside from the identification as a PGP message, the adversary could extract the ID of the recipients key, a one octet number specifying the public key algorithm used, a packet type version number and the long number that serves as input for the symmetric key recovery routine. The adversary's advantage is one.

If we look at an Academic Signature cipher of the same file in a hex editor, we see this:



```
000000006B 69 64 3A 20 61 6E 64 65 72 73 5F 32 35 36 5F 6B 31    kid: anders_256_k1
0000001220 64 6F 6D 6E 61 6D 3A 20 70 32 35 36 72 31 0A 20 42     domnam: p256r1. B
0000002478 3A 20 33 34 66 34 30 63 66 39 61 33 66 35 35 65 62    x: 34f40cf9a3f55eb
0000003631 33 36 39 35 36 39 61 33 61 35 64 62 34 30 34 34 33    1369569a3a5db40443
0000004862 66 61 36 38 34 36 31 34 38 65 66 30 63 31 33 66 66    bfa6846148ef0c13ff
0000005A32 39 38 62 31 64 39 39 66 65 64 37 30 0A 20 42 79 3A    298b1d99fed70. By:
0000006C20 35 65 37 63 65 30 39 65 38 39 30 61 66 66 66 31 32     5e7ce09e890afff12
0000007E36 64 37 33 64 63 36 66 65 38 30 33 63 62 31 39 66 63    6d73dc6fe803cb19fc
0000009034 63 37 33 32 39 37 34 39 34 30 61 33 65 64 61 64 39    4c732974940a3edad9
000000A264 63 65 62 34 62 32 65 33 35 31 20 61 6C 67 6F 3A 20    dceb4b2e351 algo:
000000B463 68 69 6D 65 72 61 20 58 3A 47 0B 2C 6F 2E F7 9F 85    chimera X:G.,o....
000000C64F E7 1E FF F0 4F 61 84 32 30 26 1A E2 D3 FA 04 CF 08    O....Oa.20&.......
000000D8D7 34 F5 D1 96 85 7D CA 24 00 70 5D 1B BF F6 45 C4 8D    .4....}.$.p]...E..
000000EA5C 47 F1 10 9A C6 F7 E0 76 27 85 A7 34 2C 62 A8 6E E1    \G......v'..4,b.n.
000000FCF1 94 4B 88 DD 1B C5 51 0C 53 C6 A1 0C 3C 2A 2B 30 CA    ..K....Q.S...<*+0.
0000010E7E 61 FC 6F F9 7E 71 FF 6B 3E 9C 84 0E B8 31 55 5F 1B    ~a.o.~q.k>....1U_.
000001209A B1 77 A2 DD 5B 75 AA 68 2A E2 D5 2F C1 86 6D CB BA    ..w..[u.h*../..m..
0000013265 30 92 C9 3E 41 81 41 97 3F 12 4D 3B C1 13 5D 5B 28    e0..>A.A.?.M;..][(
0000014432 9B AB BF FC 8A 63 2D 5D 2D D5 23 9E 8A A4 C1 C3 B3    2.....c-]-.#......
00000156F7 5C E9 EE D3 3D C0 00 B1 D3 B8 0A 07 95 A9 75 AC 4D    .\...=.........u.M
```

Signed 8 bit: 71   Signed 32 bit: 1865157447   Hexadecimal: 47

Illustration 1: Screenshot of a hex editor(GHex) showing the first bytes of an academic signature ECC-cipher

There is obviously a cleartext header in ascii that states key id, elliptic curve domain name, the two coordinates of a point "B" in the plane and the name of the symmetric algorithm "chimera". You can find the format specification at http://www.academic-signature.org/spec_ecc_cipher.pdf. End marker of this ascii header is "X:". Beyond that position, there is only nude symmetric cipher text and the remaining part thus looks like noise, if the used symmetric cipher is safe.

If an attacker intercepts a file mainly looking like noise and e.g. starting with "kid: ", he knows it is an elliptic curve cipher of the tool academic signature, he can extract the key id, domain name, basic symmetric algorithm and the point on the elliptic curve that is used in conjunction with the recipients private key to recover the basic symmetric key. The adversary's advantage is one again.

4. Dangers connected to using adversary advantage one cipher files

4.a Repudiation of encrypted communication is impossible

If you live under a repressive regime, the exposure of you being a recipient or sender of a cipher may already be enough for the regime's agencies to throw you into prison and break all your bones. If you are lucky enough to be caught in the US (but have the wrong ethnicity), your bones may remain intact but you might experience some more sophisticated torture in the form of waterboarding. You would not like that either.

The root of the problem is that you cannot deny having used cryptography.

If the repressive regime would have a remainder of rule of law and the cipher would indeed look like noise in toto(negligible adversary advantage), you might have a chance to get away with it. You could claim it were e.g. the result of last night's SETI run in the cosmic microwave spectrum. If you were really courageous, you could also call it a SITA run (Search for Intelligence in Terrestrial Agencies) but I disadvise. Anyhow, they couldn't prove you are not telling the truth.

4.b Attack path information is accessible to the adversary

Let's assume the repressive regime did not get hold of you personally, but nevertheless intercepted the message electronically. They would know immediately what procedures to try to cryptanalyze the cipher and might possibly be able to estimate, what time it would take them to get to the plaintext. Additionally they would be able to gain information on what tool you and your communication partner use and possibly what the IDs of your keys are. This would help them a lot.

Now imagine the cipher had negligible adversary advantage and you were the interceptor. You wouldn't know what cryptanalytic unit to give the presumed cipher to, you couldn't tell which cryptosystem was used, what the basic symmetric cipher was, how long it might take your cryptanalysts to regain the plaintext. You might even clog the cracking pipeline of your agency with a cipher, that is uncrackable for your organization. And, after all that, it could still just be the result of a SETI (or SITA) run. Unpleasant, isn't it?

4.c Shor's algorithm may break public key ciphers on future quantum computers

Many experts assume, practical quantum computing will be available within some decades. If so all currently established public key cryptosystems will fall victim to Shor's algorithm. If your public key is known to an adversary owning a practical quantum computer, the adversary could factor your RSA module or calculate the discrete logarithm of your public elGamal or ECC key and thus recover your private key. On a first sight, this seems to implicate the end of classical public key cryptography. All secrets of the past could be exposed if the ciphers would have been intercepted and stored.

Quantum computing, however, is useless to the adversary, if he does not know the header info given e.g. in OpenPGP or plain Academic Signature ciphers. In a negligible adversary advantage cipher file, header information cannot be extracted.

Header information, i.e. the exponentiated number in RSA or elGamal or the point "B" on the elliptic curve in ECC, is needed to regain the symmetric algorithm key on the recipient side even for the legitimate recipient. Thus a properly protected header renders classical public key cryptosystems quantum computing secure. (Symmetric algorithms are also subject to improved attacks on a practical quantum computer using Grover's algorithm. This attack, however, can easily be neutralized by using an algorithm of sufficient key size.)

4.d Cipher file contains unknown amount of information

Any information accessible to the adversary and buried in the cipher file clearly is to be avoided.

The user of a cryptographic tool is rarely aware of information buried in the cipher files, neither is this information easily accessible to the legitimate sender and recipient. They may well be IT lay people. Buried information degrades clarity and transparency for the legitimate user.

In the perfect, beautiful case of a negligible adversary advantage hybrid cipher file, the only information accessible to the sender and possibly the intercepting adversary is meta information or external information: What is the file name, who is sender and recipient, if not concealed by using a mix network, how long is the file, when was it sent, received or (possibly) intercepted. That's it, nothing is to be learnt from the cipher data itself. The amount of accessible information would be sparse, transparent, exclusively external and easily manipulable by the legitimate user.

5. How can negligible adversary advantage and quantum computing security be achieved in an encryption tool

5.a Fundamentalist approach

You may agree on a specific fixed symmetric cipher and a specific fixed procedure for salting and stretching to develop the key from a human manageable passphrase or keyword. In this case you may use the bare output of the symmetric algorithm as cipher file. Your communication partner implicitly knows that cipher files addressed to her/him adhere to these fixed procedures.

If symmetric algorithm and key preparation are properly chosen, this modus operandi of beautiful simplicity achieves negligible adversary advantage(if the symmetric cipher is sound) as well as quantum computing security, if key size and block length are sufficient.

There are drawbacks, however. There is no algorithm agility i.e. you are stuck with the selected procedures. In order to minimize the frequency of the administrative nightmare of changing the algorithm or the key derivation procedure, you'll have to incorporate an ample security margin. This ample security margin may not yet be available at definition time(any alternatives to aes?) or at least the generous security margin might come at the price of increased execution times. You could avoid this problem by coasting closer to the limit of marginal security and aid more frequent procedure updates by including procedure details as part of the symmetric key. In this case, however, you cannot exchange symmetric keys any more by conspiratively getting together with your partner, slipping a chit of edible paper with your human manageable keyword scribbled on it. You'd need to migrate to riskier behaviour and transfer the keys data chunks e.g. via exchanging thumb drives and sticking them into your respective hardware.

The biggest drawback in my opinion is the incompatibility with established public key procedures. People will not be willing to abandon the comfort of using public key procedures just for the theoretical concept of negligible adversary advantage. After the advent of practical quantum computing, the necessary change could then be abrupt and painful.

5.b Evolutionary approach

In reasonable practical public key hybrid ciphers, format specifications and access information would not be spread out all over the cipher file. Instead they would come bundled as a header or trailer of very limited size. Thus we could boost security to negligible adversary advantage and achieve quantum computing security by employing an extra layer of symmetric encryption just for this format and access information block. Please note that only a tiny amount of data will have to be protected in this way. Thus an ample security margin for the price of slower execution in this minute fraction will be easily tolerable.

There are two ways to achieve this. The needed symmetric key could be included in the public key or could be derived from public key information. In this document I will call this the intrinsic key variant. Else the symmetric key could be supplied via an additional channel. This will be called the extrinsic variant.

In the intrinsic variant, the public key can not be treated as fully public any more. It must be treated as "semi public" or confidential within a communicating group. It is a group secret and should be kept inaccessible to adversaries external to the group like e.g. the malevolent agencies NSA or GCHQ.

In the extrinsic variant, the separate symmetric cap key must be kept confidential within the communicating group and be kept secret from hostile agencies having a practical quantum computer at their disposal. Please note that in this extrinsic case, the public key may even remain visible to hostile agencies. Even with knowledge of the private key, header information would still be necessary to decipher the cipher file.

Transition to post quantum computing security would be relatively smooth in both variants.

Transition for the intrinsic NADA-Cap key variant: After the advent of practical quantum computing in hostile agencies, a new public private key pair would have to be created for each communication circle, the user is taking part in. This would be easily manageable e.g. one new semi public key could be created for your workplace, only to be shared with your colleagues and another new semi public key for your family and your closest friends. This should usually suffice. Post quantum computing authentication and privacy could no longer be securely maintained against powerful agencies for a retained public key, which had been visible publicly. Authentication and privacy could remain secure within your respective groups as long as the semi-public key remains confidential within the group i.e. no group member is a traitor, has his/her computer pwned or has a practical quantum computer at his/her disposal. Negligible adversary advantage would come as a free add-on against adversaries external to the respective groups who may be eavesdropping on the groups communication channel.

Transition for the extrinsic NADA-Cap cap key variant: The communicating group has to meet conspiratively once and agree on a NADA-Cap keyword shared among the group. Public keys could be retained and continued to be used for old pattern public key cryptography vulnerable to powerful agencies. Encryption of the access info using the NADA-Cap keyword alone would be sufficient to maintain privacy against powerful agencies, who may be eavesdropping on your communication channel. A problem would be, however, that public key authentication could not be achieved securely any more with this variant when public keys have been visible publicly once.

Please note, that for the time being the NSA recommends to add an additional AES symmetric encryption on top of public key encryption to render encryption post quantum computing secure. After having read the preceding remarks you should recognize this recommendation as being inefficient and clumsy. On top of that, it would also be insecure since AES has insufficiently long keys for security against attacks via Grover's algorithm in conjunction with a practical quantum computer.

6. Example of NADA-Capping in Academic Signature

Since version b53, published on 11/16/2015, the software tool Academic Signature allows to apply a NADA-Cap to elliptic curve hybrid ciphers. Upon checking the box "Apply NADA-Cap" the user is asked for a NADA-Cap keyword in the enciphering dialogues. If the keyword "intrinsic" is given, the NADA-Cap key is derived from public key data of the recipient. Otherwise a human manageable keyword is assumed and a 4096 bit NADA-Cap key is derived from this keyword using ample salting and stretching to ward off dictionary attacks.

Subsequently the access data in the cipher's header is encrypted using the 4096 bit key and block size algorithm "F_cnt_lc"(counter mode). The cipher file then looks like pure noise in toto. Since Academic Signature also uses PLSC(payload size camouflage), the file size is inflated by a random amount that scales with the square root of the initial plain text size. This limits the size information considerably for an intercepting adversary. I will not present a hex editor screenshot of a NADA-Cap protected cipher here. It would be boring - just a bunch of random looking data indistinguishable from true random data.

The reader is invited to check that himself. The reader should also feel encouraged to submit an arbitrary NADA-Capped Academic Signature cipher to any "Diehard Battery of Tests" of randomness of his liking and inspect the results.

On the receiving side, on the inability to extract header information from a presumed cipher file, the user is queried for a NADA-Cap keyword to be used to initiate deciphering. If the intrinsic variant was used, the recipient should know what communicating group the sender is belonging to and select his/her corresponding private key in conjunction with the NADA-Cap key "intrinsic". Otherwise the same salted and stretched derivation procedure as on the encryptor's side is used on the NADA-Cap keyword to derive the 4096 bit key. On success the header block would be uncovered and the elliptic curve cipher can be decrypted conventionally using the recipients private key.

7. Conclusion

Even though a quantum computing secure replacement of conventional asymmetric cryptosystems has not been earmarked yet for standardization and is far from being publicly, let alone freely, available in a usable form, current hybrid ciphers can be rendered quantum computing secure with the introduction of a NADA-Cap based on symmetric encryption using sufficient key size.

The presented methods do offer viable solutions for achieving privacy, but do not offer a solution yet to the problem of secure, publicly verifiable digital signatures for the time when practical quantum computing is widely available.

A NADA-Cap limits the information accessible to the adversary to minimal information purely external to the cipher file. The sparse external or meta information is easily visible and manageable by the legitimate user of the used cryptographic tool. Thus as an additional advantage of NADA-Capping, conventional cryptanalytic attempts of powerful adversaries are substantially aggravated.

NADA-Capping allows for a smooth transition from hybrid encryption vulnerable to practical quantum computing to post quantum computing secure hybrid encryption. It is available today. No new primitives are required, but some protocol changes have to be introduced. In the intrinsic variant, the concept of "semi-public" keys needs to be introduced. In the extrinsic variant, a shared symmetric key needs to be agreed on in a communicating group. The set of two access concepts "private" and "public" needs to be extended to include a third element "group internal". For the future, NADA-Capping can be regarded as backup method, in case the candidates for post quantum computing asymmetric primitives should fail or not be available in time.

There are no paper based references, because the reader can look up all the cryptographic concepts I am using here(except the NADA-Cap itself) in a good textbook on cryptography.

The author of this paper is also author/developer of the cryptographic tool Academic Signature. Academic Signature is distributed freely, based on a public license and the author has no financial interest regarding this tool.

The reader may have noticed, though, that I do have a political opinion about state sponsored agencies abusing their power to eavesdrop on all ordinary citizens and, despite prayer wheel like issued statements to the contrary, break laws, violate their respective state's constitutions and, in the authors opinion, desperately lack reasonable supervision. This opinion admittedly did diffuse into some passages of this writing, which may occur unusually explicit and partisan for a technical paper to the reader.
