# All.Net Analyst Report and Newsletter

***Welcome to our Analyst Report and Newsletter***

## Encrypt it all!

The more I read about the way governments and companies are accessing and abusing the massive quantities of information exchanged by people, the more concerned I am about where it is all going. 1984[1] may be a bit late, but it is coming in one form or another. And the only way for individuals and businesses to protect themselves is to encrypt early, often, and always. Perhaps more importantly, you should encrypt things you don't care about so it makes it harder for the subverters to pick out what's important. Don't trust others to do it for you, certainly don't trust the "cloud" to do it for you, but rather do it yourself. And don't trust the programs that do it for you or the key management systems provided by the big vendors, because they are all being subverted by the government. Don't trust the hardware devices being sold to encrypt because the hardware is being subverted from source to destination and along the way. And don't trust the algorithms approved by governments because the governments have inserted intentional weaknesses into those systems. If you are using a computer program to transform your program source code into executable code (i.e., a compiler) that may have Trojan horses contained within it, so you cannot trust that either. You might be able to trust older hardware and software that was in place before the governments started to subvert all of those systems, but how far back do you have to go? Let's say the 1980s?

## Hmmm... this may be a bit of a problem.

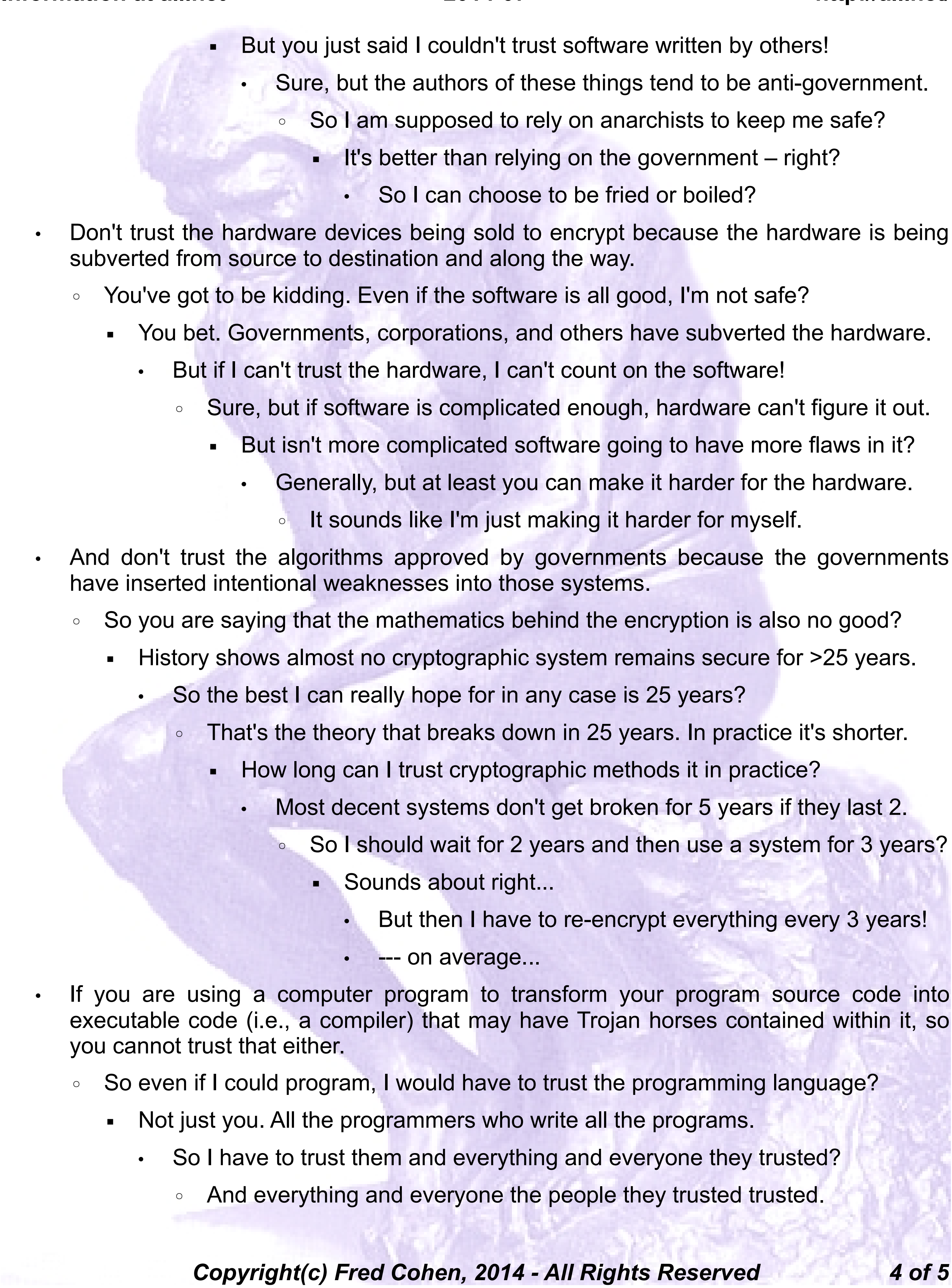
The more deeply you look into this, the trickier it gets. The notion of "encrypt it all" may sound reasonable and prudent, but when you start to look under the covers, there is a tradeoff between what you get and what you pay for it. Let's go through that again...

- The only way for individuals and businesses to protect themselves is to encrypt early, often, and always.
    - If you encrypt it on the screen, you will have problems reading it.
        - If you don't, screen scraping (common surveillance method) will work against it.
    - If you encrypt it in memory, computation may be a problem.
        - Over the last 30 years or so, some in use encryption has come to be useful, but relatively little of it. For example, you can't do email or edit a document with it.
    - If you encrypt it in storage, disk problems may lose it all..
        - And there is the problem of control over backups and encrypting them.
        - And there is the problem of keeping track of, control over, and secrecy of keys.
        - And how do you guarantee this when others store it?
    - If you encrypt it in transit, some folks will be able to break into the links.
        - In theory you might be able to get around this, but in practice it happens daily.

[1] The book by George Orwell, not the year...

- - - And if they record it, when the cryptosystem is broken later, they may read it.
      - And rest assured, almost every cryptosystem is broken eventually...
  - You cannot encrypt all the keys – or else you won't be able to access them.
    - Which means you typically protect the keys with a password.
      - But if a password is protecting the protection, how do you protect it?
        - It might be easier to protect a few things well than a lot of things...
          - But if you forget the password, you are screwed...
            - Unless there is a way around it...
              - In which case that is the weak point.
      - And if a password is good enough for the keys, why not trust it for content?
        - Perhaps the content is more subject to physical theft of media?
          - But usually the content is stored in at least as secure a place as keys
            - Except in a cloud environment where I store content but not keys
              - In which case you still have to protect the keys...
- Perhaps more importantly, you should encrypt things you don't care about so it makes it harder for the subverters to pick out what's important.
  - But encrypting everything means I need to expose the keys more often to get to it
    - Yep, so you need to protect the system that protects the content
      - But if I can protect the system that protects the content, why encrypt?
        - It may make it harder when they break in – it increases attacker workload
          - But encryption increases my workload...
            - You can't get something for nothing.
              - But if we all encrypt all the time, we are all wasting a lot of time!
                - Yep – and in that sense, the attackers have won.
          - If they break in, can't they plant a Trojan to enter later?
            - Yep.
  - Doesn't it cost more and run slower to keep everything encrypted all the time?
    - Modern hardware encryption is very inexpensive and fast, but its costs a little.
      - But isn't the real cost in managing and maintaining it all?
        - That's a big part of it – but the bigger cost is if you lose a key.
- Don't trust others to do it for you, certainly don't trust the "cloud" to do it for you...
  - But if I have to do it all myself, that means I have to run an encryption capability.

- 
  - 
    - Yes you do, but there are lots of hardware and software tools to help you.
      - So I have to buy hardware and software and learn how to use it?
        - Yes, you do. But it's not that expensive, and you can learn it quickly.
          - But if we all do this, won't it cost us all a lot and take a lot of time?
            - Sure, but it feeds the security business very well and creates jobs!
  - Why not just trust the cloud providers instead of putting all this money and time in?
    - Because they have proven themselves to be untrustworthy.
      - Not all of them...
        - Most of the biggest ones have been broken into by the NSA and others.
          - But I'm not a criminal, so they won't come after me...
            - First they came after the gypsies, but I wasn't a gypsy...
              - Isn't that just paranoia?
                - It's not paranoia if the fear is realistic and demonstrated.
- And don't trust the programs that do it for you...
  - Why not trust programmers? I trust them for the operating system and applications.
    - That's why we need encryption. Because these other things keep failing.
      - But these things that keep failing are the things that run the encryption!
        - True enough, but maybe break-ins won't be able to break the encryption.
          - Isn't the history of this that they get access to the keys?
            - Sure, but you can use special hardware to protect the keys.
              - But can I trust that hardware and the software that runs it?
                - Not really, but you have to trust someone...
  - But I'm not a programmer, and I can't really do this by hand you know.
    - If you can't write programs, you can't control your own destiny in the cyber-age.
      - But I thought encryption was going to be the cure for these problems?
        - Not really, encryption is just the start of the next set of problems.
- And don't trust the key management systems provided by the big vendors, because they are all being subverted by the government.
  - So you're saying I cannot trust the people who sell me the keys for encryption?
    - Right. So in order to be safe, you need to create and manage your own keys.
      - But I don't know how to create and manage my own keys.
        - No problem, there is software to automate this process.



- But you just said I couldn't trust software written by others!
  - Sure, but the authors of these things tend to be anti-government.
    - So I am supposed to rely on anarchists to keep me safe?
      - It's better than relying on the government – right?
        - So I can choose to be fried or boiled?

- Don't trust the hardware devices being sold to encrypt because the hardware is being subverted from source to destination and along the way.
  - You've got to be kidding. Even if the software is all good, I'm not safe?
    - You bet. Governments, corporations, and others have subverted the hardware.
      - But if I can't trust the hardware, I can't count on the software!
        - Sure, but if software is complicated enough, hardware can't figure it out.
          - But isn't more complicated software going to have more flaws in it?
            - Generally, but at least you can make it harder for the hardware.
              - It sounds like I'm just making it harder for myself.
- And don't trust the algorithms approved by governments because the governments have inserted intentional weaknesses into those systems.
  - So you are saying that the mathematics behind the encryption is also no good?
    - History shows almost no cryptographic system remains secure for >25 years.
      - So the best I can really hope for in any case is 25 years?
        - That's the theory that breaks down in 25 years. In practice it's shorter.
          - How long can I trust cryptographic methods it in practice?
            - Most decent systems don't get broken for 5 years if they last 2.
              - So I should wait for 2 years and then use a system for 3 years?
                - Sounds about right...
                  - But then I have to re-encrypt everything every 3 years!
                  - --- on average...
- If you are using a computer program to transform your program source code into executable code (i.e., a compiler) that may have Trojan horses contained within it, so you cannot trust that either.
  - So even if I could program, I would have to trust the programming language?
    - Not just you. All the programmers who write all the programs.
      - So I have to trust them and everything and everyone they trusted?
        - And everything and everyone the people they trusted trusted.

          - And so forth ad infinitum?
            - Ad finitum... there are only a finite number of people and things.
              - Fair enough – I only need to trust everyone to trust anyone...
                - Exactly.

- You might be able to trust older hardware and software that was in place before the governments started to subvert all of those systems, but how far back do you have to go? Let's say the 1980s?
  - They have been doing this since the 1980s?
    - Actually long before that, but they usually ignored high volume commercial.
      - How do I know if the one I choose is one they ignored?
        - You don't, but you could use variety in a redundant configuration.
          - So now I have to be a computer engineer as well?
            - It's a good start, but hardly all you need...

### What does this really mean?

Encryption has always had, has today, and for the foreseeable future will have, all of the same problems that other technologies have in terms of effective protection. From the basic concepts through the details of implementation and use, protection systems fail.

- If you are relying on encryption for long-term protection of high-valued information, you are likely making a mistake. Don't do it.
- If you are using encryption to make things a bit harder for some attackers some of the time, be careful you don't end up costing more than you gain. If they delete your keys, you will lose everything encrypted, possibly forever.
- If you are using encryption to gain anything but secrecy / privacy, you are likely using a hammer to try to screw in nails, and it won't work.
- If you are thinking of applying encryption across an enterprise, look carefully at the costs and consequences. It is a big complicated thing to do and it's easy to make big mistakes.
- If you didn't know and fully understand all of the things detailed above, and you didn't hire an expert who knows all of these things and the subtleties surrounding them, the encryption system you put in place is likely full of holes. You might want to start looking a bit more closely.

### Summary

Like most such things, it sounds good at first, but when you really look into it, it starts to hurt. Beware the solution to all problems. It never meets expectations. Seek rather the tradeoffs between solutions. Cryptography is useful in many ways, it raises the bar for some attackers in some circumstances, but it is limited in its effectiveness and utility. It has costs and consequences, and only works well in the context of a larger overall protective scheme.