Scale

Our cyber risk is growing exponentially and following the same kind of arc as the global population explosion and the explosion of mobile and smart devices. This makes our collective problems seem like they are bigger problems to solve.

As our vulnerability increases so does our blindness to threat. Maybe it is too much to consider when we feel helpless.

No one noticed when budgets went from being listed in billions of dollars to trillions of dollars.

No one notices as unfriendly nation states are practicing and perfecting the shooting of rockets into space to take out communications satellites. We thought our massive reliance on global telecom would be safe up there. We didn't consider that near space would become a cyber war theater. But we didn't predict or anticipate simpler things like the internet running out of addresses either.

Hackers (both legally sanctioned and not) are getting smarter. They are identifying bottlenecks on our systems and points of commonality between them and the results of cyber hacks are getting exponentially worse.

But no one seemed to notice when GeMalto, the world leader in SIM cards was hacked and over 1 billion cell phones were open to video and voice and data surveillance without warrant.

That is 1/7th of the world population!

Android and Google was recently hacked and that exposed 1 billion people to a similar vulnerability.

That is 1/7th of the world population!

How does this come to pass? One reason is that when we by necessity harmonize technologies to create the greatest leverage in communications and commerce we create an environment where we all share the same vulnerability.

And there is an inverse relationship between the seriousness of a topic and the number of people that actually understand it. This is why legitimate scientific speculation is backed up with scientific method and actual demonstration. Most people don't know how a phone works and even fewer people understand how a telecommunications framework operates but they can see the results and they can use it.

Everyone has a narrative

In cryptography everyone has a narrative and they generally work based on their own self interest and not the public or collective interest. And their self interests are usually at cross purposes. This is exactly why the balancing of security and privacy (and keeping it flexible) is so difficult.

- Governments and law enforcement have legitimate concerns for national security
- Corporations and citizens have legitimate expectations for and concerns about privacy
- Many corporations game standards regimes to create compelled markets for their own substandard technologies. They also use them to create barriers to entry to new technologies and competitors.
- The Internet enables those without credentials to write anything to advance their careers

This all creates a cacophony of noise on the Internet that is challenging to sift through and determine what's accurate and what's not. The Internet seems designed to confuse consumers and citizens.

This paper is an attempt to cut through this white noise to help the uninitiated in understanding information about cyber security in general and cryptography in particular. To do so we are going to do an analysis of a case study looking at the security of an algorithm when deployed in a microprocessor in an effort to either prevent side channel attacks based on power analysis or more importantly to dramatically improve the security of information technology processes and components that use counters, circular shift registers, line feed shift registers, key rings etc. This is a critical problem because there are countless instances of their use and their security is essentially non-existent.

Preventing side channel attacks on these kinds of components is historic because it has never been accomplished before.

First things first – analysis of an algorithm versus an analysis of an implementation

Algorithm security

All cryptographic algorithms and processes undergo two specific kinds of analyses.

A security analysis evaluates whether there are any known techniques, mathematical or otherwise, that would make the cipher, the essential building block of all crypto security systems, breakable or vulnerable just because of how it is constructed.

Generally these evaluations are done on software implementations. Scientific papers and scientific data, write-ups, patent descriptions, and **demonstrable test results** all become part of the evaluation. An algorithm can be deployed most easily in software and testing is usually done against a software deployment. Software also has the least physical impact and minimizes the introduction of unintended variables that occurs when physical devices or components are actually physically changed to try to process that algorithm as a hardware driven function as opposed to an exclusive software function. As an example look at Intel's AES NI (new instructions) where they make the chip do more of the harder mathematical calculations because the performance of those calculations is so abysmal with negative consequences when done in software.

A performance analysis evaluates metrics we can measure on various computers or devices or components like speed, randomness, resistance to brute force attacks, the amount of computational effort required, the number of processes required, and the amount of memory or storage required.

At this level the proper comparison is between ciphers so an appropriate comparison is looking at the performance of the Whitenoise dynamic cipher and comparing that to AES, DES, RC4 and other ciphers that generate keys.

DES was considered a secure algorithm until a paper was published demonstrating the correlation between the amount of money spent on the resources required to break DES and the amount of time to do so. DES was the first AES (Advanced Encryption Standard) sanctioned encryption algorithm. It was intended that the algorithm would be robust enough to last 70 years. They thought that the necessary advancements in computers to attack it would take that long. It was intended that DES would be used for only 35 years out of extreme caution. DES survived only a few years because it was so weak.

The current standard is AES (Rijndael). At least empirically we know it is not secure enough because of its key sizes, randomness and the current speed of computers. The United Kingdom and the United States recently indicated their belief that Russia and China had decrypted the Snowden files, presumably without any of the keys, by repatriating personnel and assets that would be endangered by their disclosure.

Given the massive body of attack material against AES security particularly when implemented with RSA style public key infrastructures it is questionable whether that cipher would survive a security analysis now because of lack of randomness and the current speeds of computers. We know that Microsoft and PGP have moved away from NIST recommended random number generators because they don't create random enough data to be used in cryptography which requires a secure data source.

Whitenoise is the most secure cipher and deterministic random generator known on all metrics. It is strong, fast and dynamic. It operates as a true one-time-pad which is why it has been patented globally. A one-time pad is the only provably unbreakable cipher.

The security analysis by the University of California, Berkeley could find no mathematical technique or any other technique with which they could break Whitenoise. They said, 'if there was a magic computer that could do a trillion-trillion calculations per second, and if there were a trillion-trillion of these computers spread through out the universe, and if you ran those computers for a trillion-trillion years that the odds of breaking a Whitenoise key would be ½ ^{1350} which is unimaginably small.'

Similarly, the performance analysis by the University of Victoria, British Columbia ECE (also paid for by the National Research Council of Canada) had historic, landmark results. They tested Whitenoise against the NIST (US National Institute of Standards and Technology) test suite that was configured to be an order of magnitude more sensitive than what AES was tested against. The tests allowed only one statistical failure for every thousand rounds instead of allowing for one statistical failure for every hundred rounds.

Not only was Whitenoise the first algorithm to generate sufficiently random data on the first round, it did not generate a single statistical failure in weeks and weeks of testing against a supercomputer array.

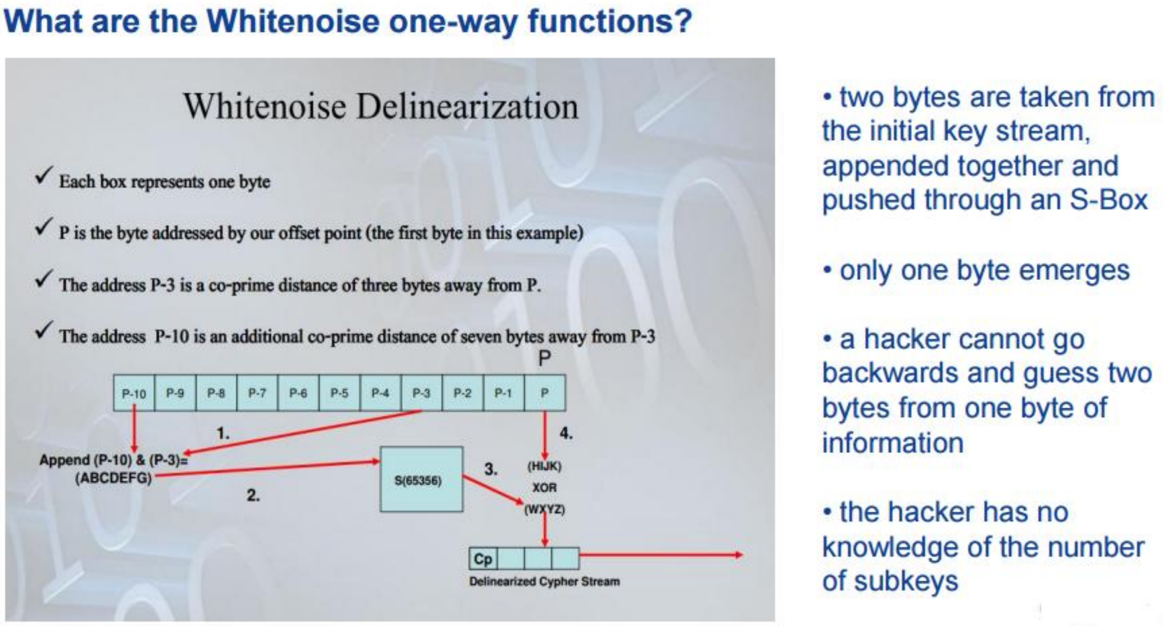
--

Central to the security and robustness of cryptographic algorithms are the one-way functions it deploys.

One-way functions are things that are supposed to be easy to do in one direction but considered impossible to do in reverse.

The standard ciphers develop one time functions with mathematical functions. That creates problems. If you can construct something mathematically you can deconstruct it mathematically.

Whitenoise one-way functions are essentially mechanical and therefore are not vulnerable to any mathematical attacks.



What are the Whitenoise one-way functions?

Whitenoise Delinearization

✓ Each box represents one byte

✓ P is the byte addressed by our offset point (the first byte in this example)

✓ The address P-3 is a co-prime distance of three bytes away from P.

✓ The address P-10 is an additional co-prime distance of seven bytes away from P-3

- two bytes are taken from the initial key stream, appended together and pushed through an S-Box
- only one byte emerges
- a hacker cannot go backwards and guess two bytes from one byte of information
- the hacker has no knowledge of the number of subkeys

Finally, the hacker has NO KNOWLEDGE of the master key which is used to populate the variable subkey lengths.

The absolute robustness of the Whitenoise algorithm has been affirmed over and over and over and over. It’s been recognized by the White House, the United Nations ITU, international standards groups, international security awards, unchallenged in public contests and successfully patented internationally.

Whitenoise is secure.

Implementation security

Implementation of cryptographic based network security controls requires a framework, a means, of accomplishing the **three essential tasks** of a crypto system: key creation, key distribution and key management.

There are only two prevalent approaches to the key distribution challenge that create two distinct frameworks.

RSA-style public key infrastructures (PKI) are asymmetric frameworks. It requires that there are public keys (always accessible) and a complex, highly intensive mathematical handshake to generate the keys and share them between sender and receiver. And yet, it is always, and has always been, vulnerable to man in the middle attacks. And over the decades it has become breakable by a host of other techniques.

Dynamic distributed key infrastructures are symmetric, distributed key systems where the server and the endpoint have an identical copy of a user or endpoint's unique private key. That key is distributed one time. Thereafter, this key can create an infinite number of keys for unique accounts and each of those keys can create an infinite number of one-time-pads. The single distributed key performs all network security controls as a dynamic one-time-pad without ever exchanging key or offset material again. As such it is secure. Please review:

Comparison of asymmetric PKI and symmetric DDKI handshakes

Implementation security analyses are addressing a different set of problems on how to use cryptographic ciphers to solve various computer and communications security and identity problems.

Any conclusions that might be derived from investigations on various implementations do not challenge the underlying algorithm.

Implementation research is looking to learn how to improve the way those algorithms are deployed to solve specific problems. And we don't want the implementation to affect the crypto's inherent security characteristics or introduce more problems. The goal is to learn the most effective way to attack the most pressing security challenges. Too often we see the misuse of this process where persons are determined to advance their own inaccurate narrative and disregard scientific method and rigor.

Let's look at a few different contexts that Whitenoise has been deployed in and the security problem we were solving.

- Single key banking where only the bank has the key
- Radio Frequency Identification, sensors, Internet of Things and environments that are cost sensitive for manufacturing and/or have incredibly limited storage, power, and processing capabilities.

- Cloud deployments that allow endpoint to encrypt and upload any volume of data so that the cloud provider never has their key.