# MySQIF™ Privacy App™ augments an organization's internal security

One of the most frequently asked questions about MySQIF<sup>™</sup> Privacy App<sup>™</sup> by network security engineers is: Why do I need it since I already have encryption in my network security (like Crowdstrike!)?

Leaving the issue of the vulnerabilities of particular encryption protocols aside, MySQIF<sup>™</sup> provides a critical new element to a secure internal network.

MySQIF<sup>™</sup> mathematicians and engineers have advised and supported many of the world's largest networks. They have applied that knowledge to the



These "trusted" networks just SHUT YOU DOWN!

When poorly-tested Crowdstrike-Microsoft updates shut your network down, your key personnel file sharing was exposed to predators as engineers scrambled for alternative ways to communicate and share fixes. This cost some tens of millions.

MySQIF<sup>™</sup> Privacy App<sup>™</sup> gives your people an unprecedented ability to continue sharing your vital files away from prying eyes who exploit such crises to sneak in and steal your remediation communications to steal secret documents and network fix configurations for future attacks.

SHARE FILES WITH CONFIDENCE USING MYSQIF™ PRIVACY APP™

engineering of MySQIF<sup>™</sup> Privacy App<sup>™</sup> that uses powerful polymorphic mathematics to make its encryption impenetrable using one-time, unique key fingerprints.





Try **MySQIF™ Privacy App™** for seven days free, no credit card needed until you register.



For more information, go to mysqif.com

Figure 1: MySQIF<sup>™</sup> Privacy App<sup>™</sup> helps keep your network secure during a breach event.



All organizations experience network security breaches. Breaches are a fact of life. Breaches can allow an attacker unrestricted access to the organization's communications until the breach is discovered and repaired.

## For individual users, if companies with big information technology budgets cannot protect themselves, how does an individual user stand a chance against predators?

Individual users are less likely to be targeted because attackers view businesses as big scores. But, there are attackers who bet on large numbers of individuals over small numbers of businesses, precisely because individuals have fewer resources to spend on security. Once communication has left the business network the business loses most of the control over its security... until that is, MySQIF<sup>™</sup> Privacy App<sup>™</sup> emerged.

**Files sent using MySQIF<sup>™</sup> Privacy App<sup>™</sup> remains secure even during a breach.** This means that during a breach event, the organization will still have a secure mode of communications within the network. See Figure 1 above.

Whether or not the organization's data is valuable ("I have nothing to hide"), it is nobody's business according to human morality as well as the Fourth Amendment nonetheless.  $MySQIF^{M}$  can help maintain a significant degree of secure communications even *during* a breach.

During a breach event, MySQIF<sup>™</sup> Privacy App<sup>™</sup> can give engineers more time to determine the best solution. Normally during a breach security engineers are under the gun to close the breach quickly. A hasty fix often inadvertently opens new vulnerabilities for attackers.

Engineers working on the breach need to communicate and collaborate to resolve the problem. Obviously, they cannot communicate using their network until the breach is fixed. MySQIF<sup>™</sup> Privacy App<sup>™</sup> gives them a means of communications where the attacker cannot listen in. This means attackers are less likely to know what solutions are being considered and implemented, making it far less likely that they can follow up with another successful attack after the breach is closed.



Figure 2: MySQIF<sup>™</sup> Privacy App<sup>™</sup> can protect the senior management discussions on recovery plans in order to prevent a follow on breach from intercepting those discussions by the attacker.



## What is so special about MySQIF<sup>™</sup> Privacy App<sup>™</sup>'s polymorphic encryption approach? Why haven't others implemented it before now?

To answer this question, we need to compare the MySQIF<sup>™</sup> approach to the current "gold standard" named Advanced Encryption Standard (AES). AES is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001. AES has been adopted by the U.S. government. It supersedes the Data Encryption Standard (DES), which was published in 1977. Since most information technology vendors do business with the U.S. government, they are forced by default to use AES in order to get contracts.

All encryption or cryptography makes readable data random. Then, that random data can only be reversed back into the original readable data by having the encryption key.

For example:

Readable data:	→	Encrypted version:
May God bless you and keep you.		qvC+JOkJhhmxi2cNoHxBUBQFzTTM4dO0lxFvMCp1P GY=

The magic is the *randomness* of the encrypted version.

A truly random number is essentially not crackable. By contrast, an encrypted file is crackable if it was created with pseudo-random numbers. This means some characteristic exists in the outputs that make it predictable. Once that attribute is known (the key), the item can be decrypted. Most pseudo-random number generators are crackable.

To be uncrackable, the random number generator must satisfy two requirements:

- 1. The next-bit test
- 2. State extension compromise resistance

## What does this mean in English? Try this:

#### 1. The next-bit test:

Even if 999 out of 1000 bits are known, the last bit can prevent decryption. Actually, AES does this very well, as does MySQIF<sup>™</sup>.

Pseudo-random number generators always produce patterns. For example, one might discover in the encrypted file that in every 256 characters a question mark (?) occurs. This means the pseudo-random number generator is using *one formula* to produce that question mark with *predictability* every 256 characters. This narrows down the problem to cracking 256 characters instead of, say a 10 MB file. Once

MySQIF Privacy App- augments an organization's internal security

those 256 characters are cracked the entire file can be decrypted without knowing the original encryption key.

*A random number generator passes the next-bit test if* an attacker can obtain a very large amount of output and still not be able to identify patterns that allow guessing future output with better than even odds. This way, even if an attacker manages to figure out part of the original message, this knowledge does not help crack the rest of the message.

## 2. "State extension compromise resistance:"

Even if an attacker knows the inner workings of the MySQIF<sup>™</sup> PRNG (Polymorphic Random Number Generator), he or she cannot simply run it in reverse to decrypt a previously encrypted file.

The MySQIF<sup>™</sup> PRNG creates hundreds, thousands, even tens of thousands of component-keys used to decrypt a single file. Run in reverse, an attacker cannot know which key in this plethora of one-time keys is used to encrypt each bit. Thus, each step backwards to crack the next bit requires an exponentially larger set of choices: 1, 2, 4, 16, 192, 36864, 1358954496, 1.85e18...

AES is not resistant to state compromise extension on its own, because it is predictable. For any given input there is a single unique output, and vice versa. Thus if an attacker can crack any step in the encryption, the attacker can run it backwards to get all previous steps. This is the vulnerability that was leveraged to crack all of the AES modes in 2013 by a MySQIF<sup>™</sup> principal and published at DEFCON.

The MySQIF<sup>™</sup> PRNG includes functions that produce a single encrypted file using a packaged array of component-keys that are unique to each encryption (created and used one time). This makes it nearly impossible to crack a MySQIF<sup>™</sup>-encrypted file since an attacker is left to pick from thousands or tens of thousands of needles in a haystack for each choice. Thus, each step backwards to crack the next bit requires an exponentially larger set of choices (new needles): 1; 2; 4; 16; 192; 36,864; 1,358,954,496; 1.85e18... Take this number out to three lifetimes of the universe and there you have MySQIF<sup>™</sup> Privacy App<sup>™</sup> standing tall.

By contrast, AES is not resistant to state compromise extension on its own, because it is predictable (deterministically reversible). For any given input there is a single unique output, and vice versa. Thus if an attacker can crack any step in the AES encryption, the attacker can run it backwards to get all previous steps. This is the vulnerability that was leveraged to crack all of the AES modes. *This is a fundamental vulnerability of the PKI (Public Key Infrastructure) approach.* 

Put another way, once one AES RNG bit is cracked, all others in that file can be decrypted with ease.

## Encryption has been a banker-usury-spypropaganda battleground since Babylon

In addition, an AES shortcoming involves a widelypublished set of pseudo-random number generators fronted



Leader<sup>®</sup> Technologies<sup>™</sup>

by an algorithm known as the Dual\_EC\_DRBG algorithm—a cryptographic or "kleptographic" backdoor advantageous to those who know about it—British & Commonwealth GCHQ and American NSA—and no one else.

The highly classified program to crack encryption of online communications and data was developed jointly by British and American intelligence programs named BULLRUN and EDGEHILL. Edward Snowden revealed that British BULLRUN had a \$800 million per year budget.

Access to BULLRUN was limited to a group of top personnel at Five Eyes (FVEY)—the U.S. NSA and the signals intelligence agencies of the United Kingdom (GCHQ), Canada (CSE), Australia (ASD), and New Zealand (GCSB).

This Five Eyes conspiracy was of the highest secrecy from its formation on <u>Mar. 05, 1946</u> until its first disclosure on Apr. 8, 2010—64 years later. The British still black out most of their peers, knights, and aristocrats who led the setup of Five Eyes. *See <u>Mar. 11, 1946, PDF p. 11</u>*.





their Babylonian Radknight merchant-bankers in the City of London since 1902. **Their** goal is to annex American back into the British Empire.<sup>1</sup>

Bankrolled by the Carnegie Institute in 1993 under Bill Clinton, GCHQ and NSA *ordered* all their technology companies to add the Dual\_EC-DRBG algorithm to all hardware, software, and firmware. It is a pseudo-random RNG that enabled an *intentional* side channel attack on all vendor software to make all software hackable—a universal kleptographic backdoor. By 2014, the NSA was forced to say that they are no longer requiring it, but few believe them. A duck by any other name is still a duck. Government never stops anything it starts, it just changes the flavor.

## **Backdoors Я Us**

## "Five Eyes" MI6 – MI5 – GCHQ - CIA – FBI - NSA surveillance compromises the sovereignty of each state in this treasonous merchant-banker secrecy, and has done since March 5, 1946.

The bottom line is that reliable encryption requires secure random number generation, which has not been the case since our spy agencies have led the drive for a kleptocracy.

The NSA required technology vendors to embed the encryption-cracking Dual\_EC\_DRBG algorithm into all hardware, software, and firmware that contained a flaw known only to the NSA. That flaw can be exploited at any time to decrypt AES-encrypted files—it is a universal back door. They claim it has been replaced, but that claim appears dubious since NIST (National Institute of Standards and Technology) registration is still being required

By contrast, MySQIF<sup>™</sup> PRNG contains no such predictability, and therefore, is exponentially safer. No "backdoor key" is possible with MySQIF<sup>™</sup> hence this is probably why polymorphic mathematics has not been used in encryption prior to MySQIF.<sup>™</sup>

Why haven't others implemented polymorphic encryption? You'll have to ask them! As with everything, someone had to be the first to invent. This is true both for our MySQIF<sup>™</sup>-encryption algorithm and for our key swapping mechanics.

The key swapping mechanics is difficult to engineer, which might be another part of the reason no one has tried it. What we did at MySQIF<sup>™</sup> to program such an environment is not trivial, but with Michael McKibben leading the charge—the inventor of social networking—we did it along with world-class mathematicians and engineers.

We have additional uniqueness for how we harden our state extension compromise resistance, but we do not wish to reveal them. Even if they are known, it will not matter, but we choose to keep attackers guessing.

<sup>&</sup>lt;sup>1</sup> Editor: (May 13, 1913). Would Annex America [says Lillian Scott Troy], p. 20. The Commercial Appeal (Memphis, Tennessee).



## AN IMPORTANT RECOMMENDATION TO FELLOW NETWORK SECURITY SPECIALISTS

## QUESTION: I am a computer network specialist. I have told my management that we are using state-of-the-art encryption. How do I introduce MySQIF<sup>™</sup> Privacy App<sup>™</sup> without destroying my credibility when I am supposed to be the expert?!

Be honest. Do not pretend to know what you do not know.

**First, encryption algorithms don't last forever.** They are only secure and "state-of-the-art" until someone figures out how to break them. Explain this to your management, and then tell them that you have learned that AES was broken in 2013 by MySQIF<sup>™</sup> mathematician professor Albert Carlson, Ph.D. and published at DEFCON that year. Dr. Carlson is a highly-consulted encryption expert by IEEE.

**Second, never claim to know something you do not know.** Did you ever really know that AES (or whatever other encryption you are using) is secure, or did you just trust others who told you it was? Cite sources instead of claiming authority you do not actually have. The NSA said AES was good. If you say that, then when it is broken, the NSA was wrong, not you. Don't claim to be a cryptography expert when you are not. You can be a computer security expert while still admitting that you do not know the deep mathematics behind encryption and have to consult the mathematics and cryptography experts for that.

If you are honest about your expertise, you will never end up in an embarrassing situation like that. Granted, the entire network security community is making dubious claims to be "secure," un-hackable," "trusted," and "safe." For example, here are examples of the plethora of online claims:



## We realize that in an environment of systemic deception, it is difficult to be the odd man out.

Once your company puts out sales ads that are not true, what do you do? Go along, correct the pitch, or add tools to your tool box?

This is the general state of the network security business today. We get it. Honesty is often a lonely row to hoe. We propose a third way to handle this circumstance below, one that uses MySQIF<sup>™</sup> as a complement rather than a competitor to the encryption that you already have implemented.



Leader<sup>®</sup> Technologies<sup>™</sup>

**Third, come to your management with solutions.** If you are the computer security expert, and you discover a flaw, work out the solution, and then approach management.

If you follow these rules, instead of an embarrassing conversation that harms your credibility, you will identifying a problem you cannot reasonably have been expected to know about previously, and you demonstrate that you are proactive and *really are* the security expert you claim to be, because you already have found a solution to the problem.

Say something like "Hey, you should know that someone recently figured out how to break AES, which we are using for a lot of our encryption. I have taken the liberty of looking for suitable alternatives that will improve our security. It is going to cost a bit, but it will give us better security than we had even before AES was broken."

If management responds with: "But you said it was secure," you can tell them that "it *was* secure, according the NSA and other major organizations, until it was broken. That is just how encryption works. Our AES is still reasonably secure until someone breaks our implementation. If we are lucky, the hacker who breaks it will not be malicious, and we will have time to close that hole with an upgrade before a new attack starts, one that might be malicious."

If the computer sending a message is compromised, no system can stop that message from being read. In that case, the attacker is acting as the sender and he/she can see the original contents. No system, including MySQIF<sup>™</sup>, can protect against that circumstance. That said, MySQIF<sup>™</sup> makes this unlikely since such predator activity can be discovered and deleted

during the key exchange before damage is done.

In addition, MySQIF™ can protect

**against** *insider* **bad apple attacks** where network-level security has more difficulty protecting, especially during a breach. Generally, while any employee can use intercepted credentials, it will more likely be an information technology employee who knows better how to bypass network access logs and intercept sensitive communications between senior managers.

If the senior managers are using MySQIF<sup>™</sup> Privacy App<sup>™</sup> to exchange files and messages, insider attackers would have to attack their individual workstations directly to gain access to those communications, rather than intercepting network traffic. Further, if those MySQIF<sup>™</sup> communications are stored on removable USB storage media during the breach, then **the possibility of interception drops to near zero**.



Figure 3: MySQIF<sup>™</sup> Privacy App<sup>™</sup> can be an important tool in your network security toolbox while recovering from a breach event.



**In conclusion**, MySQIF<sup>™</sup> Privacy App<sup>™</sup> is an important tool in the network security toolbox to protect sending communications over the Internet where network engineers cannot rely on the internal networking securities to protect messages to and from the "wild" Internet.

The strong polymorphic security of MySQIF<sup>™</sup> remains even outside of the organization's network.

### About MySQIF<sup>™</sup> Privacy App<sup>™</sup>

MySQIF<sup>™</sup> Privacy App<sup>™</sup> uses polymorphic encryption that generates tens of thousands of one-time, unclonable keys for a single encrypted file. These one-time key clusters or "shards" are based upon unique hardware fingerprints on the sending and receiving devices.

MySQIF<sup>™</sup> Privacy App<sup>™</sup> keys dissolve after one-time use. Hypothetically, in the unlikely event that the master key was somehow intercepted in the *split-second* it exists, decryption can only occur on the sender's and receiver's devices.

Mathematically, "average" encryption being cracked on a powerful computer takes about 10 to the 300th power in seconds -- just 5-10 minutes at best. AES-256, the current gold standard, takes several seconds.

By comparison, a MySQIF<sup>™</sup>-encrypted file takes 10 to the 1500th power seconds --three lifetimes of the universe. The way MySQIF<sup>™</sup> works mathematically, backdoors are impossible.

MySQIF<sup>™</sup> also grants you a legal Leader<sup>®</sup> license to use social networking that was stolen from Leader<sup>®</sup> Technologies, on all your devices. *See* <u>Legal Social</u>. This license is for paying MySQIF<sup>™</sup> subscribers only. Your message to the world can then be: "I am enjoying the use of social networking legally!"

Author: Michael McKibben, CEO, MySQIF™, Leader® Technologies™

For more information, contact:

MySQIF<sup>™</sup> Customer Service <u>https://www.mysqif.com</u> (614) 890-1986 customerservice@leader.com

#### **TECHNICAL SPECIFICATIONS for MySQIF™ Privacy App™**

- Sign Up at <u>https://www.mysqif.com</u>
- Microsoft Windows 10 (Build 19041) and higher. Requires

Windows Updates to be current and WinAppRuntime.Singleton, .Main.1.5, Webp Image Extensions, App Installer installed from the Microsoft Store

- COMODO Code Signing Certificate
- Runs on **Apple Mac** with **Parallels** (Windows virtual machine) installed





- Runs on Linux with Virtual Box by ORACLE (Windows virtual machine) installed
- Before installing, either approve MySQIF.com, or temporarily disable malware, anti-virus, firewalls, VPN, Windows Defender, McAfee, Norton, Symantec, etc. so that MySQIF<sup>™</sup> Privacy App<sup>™</sup> will be able to be downloaded and installed successfully. Once installed, turn back on any disabled security applications
- Maximum file size: 4 Gigabyte in this version. Encrypting a
- Subscription options: Yearly (\$120), Quarterly (\$40), Monthly (\$15), Decrypt-only | seven days free, no credit card required during trial
- Enterprise site license possible
- Full set of technical <u>citations</u>:
- Selected citations:
  - <u>Albert-Carlson et al. (Dec. 01-04, 2021</u>). Evaluating True Cryptographic Key Space Size. IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 10.1109/UEMCON53757.2021.9666530. IEEE.
  - NEW: <u>Mandeep Singh, Albert Carlson. (May 31, 2024)</u>. An Approach to Watermarking Using Polymorphic Algorithms For Increased Data Security. Austin Community College/IEEE.
  - McKibben et al. (Nov. 21, 2006). U.S. Patent No. 7,139,761, Dynamic Association of Electronic Stored Information with Iterative Workflow Changes. USPTO. Source: <u>https://image-ppubs.uspto.gov/dirsearch-public/print/downloadPdf/7139761</u>
  - McKibben et al. (Jun. 05, 2012). U.S. Patent No. 8,195,714, Context Instantiated Application Protocol. USPTO. Source: <u>https://image-ppubs.uspto.gov/dirsearch-public/print/downloadPdf/8195714</u>
  - <u>McKibben et al. (Jun. 05, 2012)</u>. U.S. Patent No. 7,925,246, *Radio/Telephony Interoperability System*. USPTO. Source: <u>https://image-ppubs.uspto.gov/dirsearch-public/print/downloadPdf/7925246</u>
  - FIRST AMENDED MILLER ACTNOTICE. (Apr. 25, 2019). Leader Technologies, Inc. to the U.S. Executive pursuant to 40 U.S.C. §3131 et seq. Leader Technologies, Inc.
  - <u>Petition for Writ of Certiorari. (Nov. 16, 2012)</u>. *Leader Technologies Inc. v. Facebook Inc.*, No-12-617. U.S. Supreme Court.

MySQIF-Privacy App- Quantum-proof Encryption	
Try <b>MySQIF™ Privacy App™</b> for	
needed until you register.	
SIGN UP	
For more information, go to	
<u>mysqif.com</u>	