OpenCIP

Concept Whitepaper

BETTERGRIDS.ORG

May 14, 2021

Executive Summary

Audience

This paper is oriented to energy industry leaders and policymakers responsible for addressing emergent cyber-security risks related to inter-enterprise data exchange. It assumes a basic working knowledge of external data communications practices, technologies, terminology, risks, and compliance reporting obligations. Suggested readers include utility officers and technical architects responsible for information protection, cyber-security, operating risks, and compliance. Additionally, we seek support from national regulatory standards and policymakers at NERC, FERC, NIST, IEC, IEEE, DOE, NARUC, Electric Reliability Organization (ERO) levels.

Problem

Current industry communications rely heavily on 25-year-old technologies like SSL for secure data and file exchange. SSL is fine for simple client-server or point-to-point integration. Still, it doesn't easily adapt or scale to modern distributed computing and workforce scenarios involving data transfers through untrusted networks, email systems, file shares, mobile devices, and (increasingly) third-party cloud-hosted services. To address this emerging challenge, other cyber-sensitive industries like finance, healthcare, and defense have adopted 'End-to-End Encryption' (E2EE) as their gold standard for data security and communications privacy.

Proposal

OpenCIP is a concept for energy industry E2EE developed during a recent DOE ARPA-E sponsored research & development project focused on Secure Grid Data Exchange. It is envisioned as a new open standard for transmitting NERC CIP classified, commercially sensitive, and personally identifiable utility data over untrusted clouds. It would integrate a set of modern technologies into a cohesive specification and open reference implementation to simplify deployment, reduce risk, ensure compliance, and enhance industry interoperability across various business use cases.

Benefits

OpenCIP would keep sensitive information completely encrypted and opaque to any untrusted intermediary network communications, storage systems, or cloud services. It could significantly lower implementation and operating costs by reducing the need to harden and monitor every hop in every communication path, for every protocol, for every integration. It could help improve compliance with current and draft NERC CIP Standards (2,3,4,5,6,7,11,12), recommendations from the Smart Grid Cybersecurity Committee (NIST 7628), and the best practices of other cyber-sensitive industries. And it will ultimately help facilitate an easier industry adoption and migration to cyber-safe, cloud-based distributed computing, and workforce solutions.

Background

Current State

Historically, the utility industry relied on segregated 'Operations Technology' (OT) networks for mission-critical systems and data communications. Over time, modern firewalls and network switching technologies have allowed these OT networks to be safely integrated with utility 'Information Technology' (IT) business networks and even the external internet. But moving sensitive data anywhere outside of the trusted OT network domains requires special handling to keep it safe.

The current industry standard is to encrypt sensitive data while 'in motion' and 'at rest.' 'In motion' encryption is usually based on a 'Transport Layer Security' (TLS) implementation like Secure Socket Layer (SSL). 'At rest' encryption is usually based on a 'Storage Layer Security' (SLS) implementation at the database, file system, or storage hardware level. Transport and storage layer technologies are completely different, using different algorithms and cryptographic keys. Data must be continuously decrypted and re-encrypted at the 'seams' between every data transfer and storage step.

In common practice, most integration over the internet relies exclusively on TLS/SSL. SSL support is already built into most modern software for secure web browsing (HTTP/S), file transfer (FTP/S, SFTP), and email (SMTP/S). But it is also a 25 year-old technology, originally designed for secure client-server and point-to-point integration between *two* systems, like between your browser and your bank. It doesn't easily adapt or scale to modern n-tier, or distributed computing architectures involving several data communication hops through untrusted networks, firewall gateways, load balancers, proxy servers, file systems, or (increasingly) third-party cloud-hosted services. Every hop in the integration chain must be independently secured through separate SSL connections, and every intermediary waypoint or seam is a potential exposure of raw unencrypted data. In complex distributed computing scenarios, this is very inefficient and only as secure as the weakest link.

Emerging Challenges

The utility industry is changing in ways that further aggravate the data security problem.

Increasing cyber-security concerns are prompting utilities to move more of their grid management systems into secure OT networks. Until recently, this was only necessary for transmission scale energy management systems (EMS). But the emerging best practice is to also put newer distribution management systems (ADMS, DERMS) into more secure network domains. Unfortunately, these newer systems inherently require more integrations to business systems (CIS, GIS, WFM, MDMS) that are not practical to move. Thus, data must constantly be moved between OT and IT networks.

In parallel, modern grid operations are becoming more broadly distributed. The energy value chain is disaggregating into distinct transmission, distribution, generation, retailer, aggregator, and prosumer entities. Third-party services are increasingly used for supporting operational functions like weather

forecasting, field services, and analytics. Behind-the-meter energy resources, sensors, controls, and other 'Internet of Things' devices are becoming more important to utilities for grid-edge situational awareness and non-wires alternative grid services. The modern utility workforce is also more distributed and mobile, using a variety of devices to access sensitive systems and data from untrusted locations over public clouds. Sensitive data needs to move safely *everywhere*.

Best Practices

To address these types of challenges, other cyber-sensitive industries like finance, healthcare, and defense have adopted a technical approach called 'End-to-End Encryption' (E2EE) as their gold standard for data security and communications privacy.

The basic idea of E2EE is that sensitive data is encrypted once by the data sender into a secure 'message' that can only be decrypted by the intended data recipient(s). E2EE combines the best features of data transport and storage encryption into one 'Message Layer Security' (MLS) implementation that satisfies both the 'in motion' and 'at rest' encryption requirements.

These encrypted messages can then be sent or stored using any arbitrary technology – secured or unsecured – because the embedded data is already encrypted. This is a technical concept called 'Zero-Knowledge' (ZN) – intermediary communication and storage components have zero knowledge about the contents of the message they are transmitting or storing. The ZN attribute of the E2EE approach is especially attractive for cloud-based distributed computing and public-internet communications. It may be difficult (or impossible) to know and trust all the potential intermediary hops waypoints or seams between two endpoints.

E2EE is being used in the financial industry to support credit card transaction processing compliance with PCI DSS – the Payment Card Industry Data Security Standard. Credit card numbers are encrypted directly in the point-of-sale terminal and cannot be decrypted until they arrive at the credit card company. It is also used in mobile banking, giving customers highly secure access to their account information from their phone. In healthcare, E2EE is being used to send highly personal health information between patients and providers to meet the very stringent security requirements of HIPAA. The Department of Defense rolled out its new Cybersecurity Maturity Model Certification (CMMC) program in 2020. It singles out E2EE as a critical element of their defense-in-depth strategy for sharing classified data with their 300,000 subcontractors over the cloud.

Over the last year, almost every industry has had to quickly learn how to manage a distributed workforce, with sensitive business and personal data flowing outside the traditional organizational walls. While *personal* messaging applications like Signal, Telegram, and WhatsApp have always used E2EE for privacy, these features were not originally built into *enterprise* chat and video conference tools like Teams, Zoom, or Slack. Virtually ALL of these vendors are now adding E2EE features to meet the increased market demand for more secure workforce communications.

Technical Principles

End-to-End

The most *differentiating* aspect of E2EE is that it is 'end-to-end,' where TLS/SSL is 'point-to-point. In the context of data exchange, one endpoint is usually the data sender (a system or person), and the other endpoint is the data recipient (another system or person). When only two endpoints are involved, E2EE and SSL are effectively equivalent. But once intermediary waypoints, relays, or queues are introduced, SSL becomes a more convoluted 'point-to-point-to-point-to-point…' communications scenario requiring *complete* trust of every processing node, while the E2EE approach doesn't require *any* trust of the intermediaries. Data can also be sent asynchronously or to multiple recipients with little or no additional overhead depending on its implementation, opening up a wider variety of integration patterns. In all cases, the sender must know the recipient or group in advance because the data must be encrypted by the sender so that only *intended* recipients can decipher the message.

E2EE solutions simplify data exchange over complex networks by eliminating the need to trust all the intermediary nodes and thereby reducing data leakage risks.

Encryption

The most *critical* aspect of E2EE is the 'encryption' approach. To over-simplify, encryption uses a cipher algorithm and a random key to convert 'cleartext' data into binary gibberish. The process is reversed for decryption. A poor encryption implementation renders it meaningless to a good hacker, while a robust implementation is practically unbreakable by modern computers.

The current best practice is the Advanced Encryption Standard (AES) developed by NIST in 2001. Within the AES standard, there are multiple algorithm options using different random-number key lengths (e.g. 128, 192, 256 bit) and cryptographic ciphers (e.g. CBC, CCM, GCM), which have been developed over time and will likely continue to evolve. Longer keys (256-bit) and newer ciphers (GCM - Galois/Counter Mode) are generally preferred as *more* robust, but the entire AES family is acceptable when done *correctly*.

Therein lies a challenge with using AES. It's relatively easy to do *incorrectly*. AES requires the developer to generate a good random number using a Key Derivation Function (KDF), avoid reusing the same key and Initialization Vector (IV) twice and validate result integrity with the Message Authentication Code (MAC). Additionally, both the sender and recipient must agree on the AES key length, algorithm, message format, and a secure method for storing or exchanging the secret key. Doing these things wrong (a weak key, reusing a key and IV pair, not checking the MAC, or exposing the secret key) can result in an easily compromised security architecture.

E2EE solutions can hide and automate complex AES implementation details to minimize the opportunity for developer or user mistakes that can compromise interoperability and security.

Cryptographic Keys

Cryptographic 'keys' are basically just large random numbers that are impossible to guess and impractical to find using brute force computing. Most E2EE implementations use two different sets and types of keys – *symmetric* and *asymmetric*. The AES algorithms are inherently *symmetric*, where a single key is used for both encryption and decryption. But AES doesn't address how to exchange the encryption key itself securely. This is the primary role for the *asymmetric* keys.

Asymmetric keys are *key pair sets*, where one key is used to encrypt and the other key to decrypt. They are usually much longer than symmetric keys (512, 1024, 2048, or even 4096 bits). In E2EE, the longer *asymmetric* keys are used to encrypt and decrypt the smaller *symmetric* keys, which are sent securely from the sender to receiver, either within the message or separately. SSL also uses the exact same approach during the connection setup 'handshake.'

The 'Public Key Infrastructure (PKI) is a collection of algorithms, policies, standards, and software implementations to support the creation, certification, distribution, use, and storage of asymmetric keys. PKI defines one key of an asymmetric pair set as 'Public' and the other as 'Private.' The Public key of the *recipient* is used by the sender to encrypt something. To guarantee only the recipient can decrypt it, the Private key must be kept securely by the recipient as a secret.

Similar to the problem with AES, PKI is relatively easy to use *incorrectly*. Improper Private key transfer and storage is a considerable risk. Compromised keys must be revoked immediately. A common practice is to 'rotate' (update) keys periodically to limit leakage risks – but every rotation is also a chance for key compromise. Ensuring that the intended recipient's Public key is both *authentic* and *current* is a constant challenge. Under the X.509 standard, trusted third-party Certificate Authorities (CA) must be used to 'certify' and 'sign' keys as legitimate and manage key expiration and revocation. But integrating trusted CA certificates, signatures, expirations, and revocations can also be technically complex.

E2EE solutions can hide and automate complex PKI implementation details to minimize the opportunity for developer or user mistakes that can compromise interoperability and security.

Data Validation

E2EE implementations utilize various methods to validate that the data in messages (and the provided keys) are accurate and authentic. Most of these methods rely on some form of 'Cryptographic Hash Function' (CHF), which is a fast mathematical algorithm that maps data of arbitrary size (the 'message') to a much smaller unique number (the 'hash') of known fixed size. These functions must be both deterministic and non-reversible – the same message should always map to the same hash, any small change to the message should map to a different hash, and it should

be infeasible to derive or guess the message from the hash. Commonly used hash functions include MD5, SHA-1, SHA-256, and SHA-512, which derive hash values of 16, 20, 32, and 64 bytes, respectively. Longer hashes are more robust against modern computational power, and thus shorter hash functions (MD5 and SHA-1) are no longer sufficient for many use cases. SHA-256 is now considered a minimum for high-security communications.

The 'Message Authentication Code' (MAC) of many AES encryption algorithms is just a hash value calculated from some combination of the key, message, and other parameter data. This allows the receiver to easily validate the integrity of the *decrypted* message by comparing its own calculated hash with that provided by the sender from the *original* message.

A 'Digital Signature' of that same hash is a small cryptographic proof that helps further validate the provenance and authenticity of the message. The proof involves a few additional steps by the sender to encrypt the message hash with the sender's Private key and attach this 'signature' to the encrypted message. The receiver must successfully decrypt both the message (using the receiver's Private key) and the signature (using the sender's Public key) to validate and prove authenticity. Verifying X.509 key authenticity uses a roughly similar process to validate the trusted Certificate Authority's signature of the Public key hash. It is a cryptographically robust approach that has been used for decades, but it is somewhat complex, and developer shortcuts can compromise security.

E2EE solutions can hide and automate hash calculation, encryption, decryption, and comparison steps to simplify the process of data integrity and provenance checks.

Non-Repudiation

In addition to validating message integrity and provenance, some data exchange processes also require transaction non-repudiation by an independent third party. When did the sender actually send it? When did the receiver receive it? Was the sent message exactly the same as the received message? Traditionally, this has been difficult to do using third-party intermediaries while also maintaining sensitive information secrecy. They become the 'man-in-the-middle.'

The same cryptographic hashes and signatures used for validation can also provide a simple solution for zero-knowledge non-repudiation when coupled with a central ledger. The ledger can be a simple database managed by a trusted third party or a cryptographically immutable distributed public ledger like a Blockchain. In either case, the sender and receiver only need to log their message hashes and their digital signatures to the ledger to record the entire transaction lifecycle effectively. The ledger doesn't need to see the sensitive message data to confirm that the hashes match each other, that the signatures match the hashes, and that the respective signatures of the hashes are authentic.

E2EE solutions can provide non-repudiation capability while remaining zero-knowledge.

Self-Describing

Regardless of whether a solution uses E2EE, SSL, or some other data encryption approach, encrypted data is mostly gibberish to firewalls, routers, and other network equipment. After all, that is the whole point of encryption, to hide sensitive information from untrusted eyes during transport.

But this often presents a challenge to security architects charged with protecting enterprise systems and data. Encryption makes it much more challenging to scan incoming data for malware threats and much more difficult to scan outgoing data to apply Data Classification rules and Digital Loss Prevention (DLP) policies.

However, one little-known feature of several modern AES algorithms (like GCM) allows a cleartext meta-data header to be included with the encrypted payload and validated during the integrity check. A similar but more generic approach to any algorithm is to append a custom cleartext meta-data header to the encrypted messages. Digitally signing these headers provides overall payload integrity, authenticity, and non-repudiation benefits.

Including meta-data in the message envelope allows encrypted data to be partially 'self-describing without revealing any sensitive content. The meta-data can be a simple data classification indicator or provide additional details about the content, purpose, source, destination, transaction ids, key fingerprints, digital signatures, authorization tokens, or additional data handling rules for the recipient to enforce. Commercial 'Digital Rights Management' (DRM) solutions use a similar approach to define and enforce fine-grained access controls on the recipient side to limit decryption to specific applications, time periods, IP address ranges, or content subsets.

E2EE solutions can provide non-sensitive meta-data as part of self-describing message envelopes to help monitor incoming traffic and enforce enterprise Data Classification, DLP, and DRM policies.

Integrated Compression

Some cryptographic tools can perform an optional data compression before encryption takes place. This is often necessary when sending extremely large files or file collections like an entire directory. Shrinking the file(s) first makes the encryption or decryption much faster. Alternatively, compression tools like WinZip or 7zip already include encryption as an integrated feature, although most only use a simple user-provided password which can be easily broken or compromised.

E2EE solutions can provide integrated compression of files, file collections, and directories.

Trusted Platforms

Almost all modern computers and mobile devices now include dedicated cryptographic hardware chips (usually called Trusted Platform Modules - 'TPM,' or Hardware Security Modules - 'HSM'). The operating system uses these modules to accelerate common crypto operations and securely store secrets (data and keys) in a cryptographically protected way by the actual physical device hardware.

Data cannot be moved and decrypted on another machine, the keys cannot be accessed by software or humans directly, and decryption will only occur when an authorized person is actively logged into the device. These modules also validate the state of the machine, looking for unauthorized hardware and operating system changes. They can be configured to require a separate biometric authorization every time a new crypto operation is requested. Although some standards have evolved, programming to utilize these modules is still relatively platform-specific and complex. Also, software running in virtualized server environments (and cannot thus bind to specific hardware) must use either virtualized equivalents of these modules or completely different external hardware mechanisms.

E2EE solutions can hide and automate the use of cryptographic hardware.

Solution Concept

Proposed Standard

Our OpenCIP concept is an open E2EE communication standard specification for the energy industry. It would identify and align a set of E2EE best-practice approaches and technologies for safely transmitting NERC CIP classified, commercially sensitive, and personally identifiable utility data over untrusted clouds. We believe that OpenCIP could help developers simplify deployment, reduce risk, ensure compliance, and enhance interoperability across various business use cases.

- Outline an overall reference architecture and recommend components for E2EE.
- Define a standard but extensible E2EE message payload format with integrated encryption, compression, and self-describing headers.
- Recommend a minimal (yet future proof) standard AES cipher algorithm and key size.
- Recommend a minimal (yet future proof) standard PKI key size, type, and expiration.
- Recommend a minimal (yet future proof) standard for symmetric and asymmetric key generation, storage, exchange, verification, rotation, and revocation.
- Recommend a minimal (yet future proof) standard for hashing, digital signatures, data integrity, provenance, and non-repudiation using modern cryptographic hardware.
- Recommend deployment approaches compatible with enterprise security standards, directory services, firewalls, data classification rules, DLP, and DRM policies.

Reference Implementation

In combination with the standard, we also recommend creating an OpenCIP software library as a reference implementation. This would help validate the architecture and component recommendations, demonstrate the proposed standard interfaces, and streamline industry adoption with a proven, ready-to-go implementation.

- Implement message payload, and header envelope read/write operations.
- Identify, integrate, and encapsulate open libraries for compressions, AES cipher algorithms, PKI key management, and cryptographic hardware.
- Implement lifecycle process logic for symmetric and asymmetric key generation, storage, exchange, verification, rotation, and revocation.
- Implement hashing, digital signatures, data integrity, provenance, and non-repudiation logic.
- Implement programmatic hooks for data classification, DLP, and DRM infrastructure.

Our proposed reference implementation would provide an executable program with both a command-line interface (CLI) and application programing interface (API) with compatibility to multiple operating systems (Windows, Linux) and development languages (Java, Python, PHP, C).

Potential Applications

An OpenCIP standard and software library would create opportunities to address a variety of current and emerging utility data exchange cyber-security challenges.

Secure Email

A simple but common business use case is sending sensitive files to someone outside the company via email. The current NERC recommended approach for CEII data is to use a tool like WinZip (or similar) to compress and encrypt the sensitive file with a password. The security of this approach relies heavily on the user selecting a very long and robust password, using a *different* method (other than email) to send the password to the recipient, and keeping that password secure. Anyone who intercepts or derives the password via brute force can decrypt the data. An OpenCIP solution could function like WinZip but use a PKI-based approach to remove passwords from the equation and eliminate many opportunities for human error. But an OpenCIP solution could also scan, detect, and label the file with additional content and classification meta-data to support enterprise DRM, CIP, and other auditing requirements – something that Zip files do not support. Eventually, OpenCIP could be tightly integrated directly within common email applications to make the entire process of encrypting, auditing, and decrypting entirely transparent for the user.

Secure Chat

Enterprise messaging ('chat') systems like Teams, Zoom, and Slack are replacing email as a faster and more interactive medium for workplace communications, especially in time-sensitive real-time operations. But while *personal* messaging platforms like Telegram, Signal, and WhatsApp were natively designed as secure E2EE solutions, most enterprise systems were not. Security for file attachments is slowly being added. Still, the approaches being used are neither standard nor 'zero-knowledge.' An OpenCIP solution could help address this gap by providing a standard way for operational control rooms to securely send sensitive file attachments through either internal or untrusted external messaging platforms. Content and classification meta-data would support enterprise DRM, CIP, and other auditing requirements. Again, it could be tightly integrated directly within the messaging application to make the entire process transparent to the user.

Secure File Transfer

Secure File Transfer Protocols (FTP/S, SFTP) have been the defacto standard for most utility data exchange for decades. It remains one of the few mechanisms prescribed by NERC CIP for CEII data. But the SFTP server infrastructure is expensive to maintain and must also be protected since the files are often stored unencrypted before/after transmission. OpenCIP would allow for these data exchange processes to migrate to inexpensive cloud-based file exchange services like AWS S3 or DropBox while *improving* overall end-to-end data security by ensuring the data remains encrypted throughout the process. An ideal future scenario would have the application's endpoints reading and

writing OpenCIP files directly to avoid ever having to solely rely on the file system or transport protocol to protect the data. Our proposed reference implementation would allow OpenCIP to be easily retrofitted into existing data exchange processes using either the CLI or API option. It could either supplement a legacy SFTP integration or allow for easier migration to a modern cloud-based solution.

Secure Web-Services

While web-service protocols like SOAP and REST theoretically allow for encrypted payloads, this is rarely done in practice. The various utility integration standards that use web services (IEC/CIM, Multispeak, OpenADR, SEP, OpenFMB, Green Button, etc.) are heavily reliant on SSL as the principal security layer. But the combination of SSL technical limitations and more sophisticated cyber-threats is driving increasing costs for the infrastructure necessary to scale this architecture. OpenCIP would provide the ability to safely leverage various new inexpensive and robust cloud-based options for deploying and scaling web services, including the use of Web Application Firewalls (WAF), elastic computing virtual-servers, and server-less 'lambda' gateways. Using OpenCIP for encrypted payloads, digital signatures, and non-repudiation of web-service transactions would reduce reliance on the cloud vendor and internal infrastructure to provide these basic technical services, reducing both vendor lock-in and overall solution implementation costs.

Secure IoT

Internet of Things (IoT) is a relatively new set of technologies developed for machine-to-machine communications. Amazon, Microsoft, and Google offer turnkey IoT solutions that provide all the integration services necessary to deploy, manage, and communicate with internet-enabled consumer electronics over the public cloud. Many utilities seek to integrate behind-the-meter Distributed Energy Resources (DERs) in customer buildings and homes using these IoT networks.

Within the IoT ecosystem, the Message Queuing Telemetry Transport (MQTT) has emerged as the standard protocol for data and control, similar to the role filled by SCADA in traditional utility operations networks. Many SCADA vendors in the process control industry already natively support MQTT integration as an alternative to ModBus or DNP3. And while MQTT can also be secured via SSL (MQTT/S), that only works between the device and the cloud message broker. Data transported or stored *within* the vendor IoT clouds are NOT inherently secure. OpenCIP would provide the ability to encrypt MQTT payloads sent over public IoT clouds to ensure end-to-end data confidentiality and integrity of sensitive data. This becomes especially important when using DERs for non-wires-alternative grid services since intercepting personally identifiable information, or spoofing utility control instructions are significant cyber-security risks that must be mitigated within an IoT solution.

Secure Cloud Ecosystem

The utility industry has been relatively slow to embrace cloud-based distributed computing and workforce solutions. In large part, this reluctance has been due to cyber-security concerns and a lack of clarity from standards organizations like NERC and NIST about using cloud services safely and ensuring compliance. It also requires substantial vendor due diligence and risk management controls to verify that every solution architecture component (software, servers, storage, communications, firewalls, access control, logs, etc.) is adequately hardened and auditable.

OpenCIP would allow the development of an ecosystem of interoperable *trusted* tools and services supporting the OpenCIP protocol within the utility industry. Potential solutions include –

- *Enterprise IT,* or *OT Applications* could *natively* produce or consume OpenCIP encrypted payloads, eliminating the need to process or store cleartext data elsewhere.
- *Interface Adapters* to retrofit existing legacy applications with OpenCIP compatibility.
- *Directory Services* to find and exchange public keys used in OpenCIP communications.
- *Certification Services* to verify counterparty identities and digitally sign OpenCIP keys.
- *Integration Services* to help broker OpenCIP data exchanges between communication protocols, leveraging cleartext meta-data to route and audit messages.
- *Digital Signature* and *Ledger Services* to help verify and audit OpenCIP transaction integrity, provenance, and non-repudiation.
- *Infrastructure Adapters* to help integrate OpenCIP tightly with internal directory services, access control, firewalls, data classification, DLP, and DRM frameworks.

OpenCIP would also enable utilities to safely use *untrusted* (or at least *semi-trusted*) cloud services since all data encrypted by the endpoints would be 'zero-knowledge' to all intermediary and potentially unknown third-party vendors that might be embedded parts of the solution.

Overall Benefits

In summary, we predict that the adoption of an E2EE standard like OpenCIP would provide a variety of benefits for the utility industry –

- Allow communications across public networks by ensuring 'zero-knowledge' by all intermediary and unknown network, storage, or cloud-service components.
- Significantly lower both implementation and operational costs by reducing the need to harden and monitor every hop in every communication path, for every protocol, for every integration.
- Integrate and automate best-practice features for key generation, encryption, compression, data integrity, provenance, and non-repudiation using modern cryptographic hardware.
- Remove humans and manual processes from the loop wherever possible to reduce the opportunities for mistakes or malicious actions that could compromise security.

- Help improve compliance with current NERC CIP Standards (2,3,4,5,6,7,11,12), recommendations from the Smart Grid Cybersecurity Committee (NIST 7628), and the best-practices of other cyber-sensitive industries like finance, healthcare, and defense.
- Prepare technical solutions for the next iteration of NERC CIP (and other governing regulations), which will likely include more 'zero-knowledge' type requirements for cyber-safe cloud operations and sensitive data handling.
- Improve the utility industry cyber-security maturity level and align it with the best-practices of other cyber-sensitive industries like finance, healthcare, and defense.
- Adopt proven 'horizontal' cyber-security technologies and standards that are more cross-industry and enjoy a broader ecosystem of software tools and support.
- Avoid creating bespoke 'vertical' solutions that make it challenging to integrate tools and utilize people outside the industry.
- Facilitate an easier industry adoption and migration to cyber-safe, cloud-based distributed computing, and workforce solutions.

How is it Different?

OpenCIP would address scope gaps to complement existing industry standards –

- NERC CIP doesn't recommend specific technologies, leaving it open to interpretation.
- Utility industry 'vertical' standards (IEC/CIM, Multispeak, OpenADR, SEP, OpenFMB, Green Button, etc.) are focused on specific industry semantic sub-domains and business use cases, with security left to the transport layer.
- Cyber-security 'horizontal' standards like AES, PKI, X.509, SHA-256, and SSL only address specific implementation layers and not how to combine them into a complete E2EE solution.

Industry agreement on an OpenCIP standard would provide clear guidance on E2EE for sensitive data integration that meets or exceeds NERC and NIST standards. OpenCIP is not meant to be a semantic standard for a specific sub-domain or business use case. It could be used in combination with existing protocols and approaches to provide additional end-to-end data exchange security. It would identify, integrate, and encapsulate existing cyber-security technical standards to streamline industry adoption with a proven, ready-to-go implementation.

Next Steps

Planned Activities

The primary goal of this whitepaper is to socialize OpenCIP concepts with energy industry leaders and policymakers to increase awareness of the cyber-security challenges of distributed computing and opportunities for E2EE to advance the state of the art. We believe the E2EE approach should become a best practice for sensitive industry data exchange over the cloud.

In parallel, we hope to identify and pursue opportunities to further develop OpenCIP into a formalized industry technical standard and reference implementation. This can include public funding opportunities, standards working groups, or private commercial projects, which can benefit by incorporating some of the technical mechanisms outlined.

How Can You Help?

We need help advocating the idea with national regulatory standards and policymakers at NERC, FERC, NIST, IEC, IEEE, DOE, NARUC and Electric Reliability Organization (ERO) levels. We also appreciate any direct or indirect support from utility officers and technical architects to raise awareness and identify people or forums that can advance the cause.

If you would like to get involved, please contact GridBright CTO Travis Rouillard (travis@gridbright.com), or BetterGrids President Ali Vojdani (ali.vojdani@bettergrids.com).

About Us

BETTERGRIDS.ORG

The BetterGrids Foundation is a non-profit organization supporting research and education in developing better solutions for grid optimization, control, resiliency, and integration of renewable and distributed resources. GridBright established it in 2017 for the Department of Energy. For more information, please visit our website at bettergrids.org or contact us at contact info@bettergrids.org.

GRIDBRIGHT

GridBright specializes in Secure Grid Integration. We help the electric industry implement smarter solutions for managing the electric grid. We provide integration consulting services and technology solutions to distribution utilities, grid scale developers, and researchers. For more information, please visit our website at gridbright.com or contact us at info@gridbright.com.

Acknowledgement

The OpenCIP whitepaper was authored by Travis Rouillard of GridBright as part of the ARPA-E project Secure Grid Data Exchange Project (SGDX) with input from the BetterGrids SGDX Working Group, including the following members:

- Ali Vojdani, GridBright
- Ken Anderson, ACKS Advisory services
- Michael Brown, NV Energy
- Peter Klauer, CAISO
- Ruchi Rajasekhar, MISO
- Terry Nielsen, GridBright
- Tom Williams, WECC
- Zac Canters, DataCapable
- Ziad Dassouki, ATCO