



# MEETS Security

CirQLive's Media Enhanced Education & Training Suite (MEETS) is designed by a team of security professionals with more than a decade of experience in cryptography, computer and web security. The CirQLive team has contributed security implementations, improvements, and other enhancements to major software libraries, essential to nearly every modern computing platform in use today, for example, [cURL](#) and [Qt](#)'s authentication protocols, and the [LibreSSL](#) cryptography library. CirQLive strives to adhere to, as well as exceed, industry standards when it comes to matters of security.

## Table of Contents

MEETS Security.....	1
Security of the MEETS platform.....	1
Security of entry via LTI.....	2
Role of the Learning Management System in security.....	2
Security of the connection to the LMS.....	3
Security of the connection to the Web Conference.....	3
Web-conferencing credential security.....	3
Web-conferencing privilege separation.....	4
MEETS Administration Authentication.....	5
Other Access.....	5
MEETS Environment.....	5
Development Practices.....	6
Training.....	6
MEETS in-house use.....	6
Design and Implementation.....	6
Security Research.....	7
Technology.....	7
Certifications.....	8

## Security of the MEETS platform

MEETS is accessed via [HTTPS](#) and offers [hardened Transport Layer Security v1.2](#) with [forward secrecy](#) and [elliptic curve cryptography](#) with strong [cipher suites](#) and protection against vulnerability inducing [HTTPS interception](#).

These high levels of security are currently fully supported by recent versions of Chrome, Edge, Firefox, Internet Explorer, Opera, and Safari. Browsers incapable of these high levels of security are denied access.

Various attacks against TLS are mitigated and prevented with TLS compression disabled, [Strict Transport Security](#), Secure Renegotiation, disallowing weak or

problematic algorithms, prioritizing strong [authenticated encryption](#) algorithms, and ensuring the servers' security [patches](#) are up to date.

## Security of entry via LTI

LTI is a pseudo-protocol for [federation](#) of distinct web-based services. LTI works by having an originating service provide a user's web browser some information to pass along to a destination service. The destination service can then review this information to determine where it came from, validate various authentication details, and process the information that is provided.

LTI currently mandates that entry is secured by a [shared secret](#) and HMAC-SHA1.

The [Secure Hash Algorithm](#) is [currently considered secure where collisions are not a factor](#), such as with a [Keyed-Hash for Message Authentication](#), as required by LTI. The LTI protocol utilizes [nonces](#) to eliminate attacks of the [capture-replay](#) variety. LTI currently does not officially allow for any other form of secure entry. However, MEETS itself accepts entry using the [Secure Hash Algorithm 2](#) via HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, and HMAC-SHA512 as well. If your LMS allows these stronger algorithms, enable the strongest offered.

MEETS uses a [cryptographically secure pseudorandom number generator](#) hardened with [Hedging Deployed Cryptography](#) to generate a new 384-bit secret for each instance. Since HMAC-SHA1 (or later) is used, it counters [preimage](#) and [length extension](#) attacks. Nonce uniqueness is validated to prevent possible attacks described above. The MEETS LTI endpoint is structured to limit varying output, and uses constant time algorithms wherever possible, in order to be hardened against [web-based side channel attacks](#) such as [timing attacks](#) or [differential fault analysis](#) using malicious [POST](#) messages.

However, despite the strength of the LTI protocol and the design of MEETS to avoid potential LTI security pitfalls and employ attack countermeasures, security is only as strong as its weakest link. Thus the burden of LTI entry security is upon the Learning Management Systems to provide a secure environment to a user's web browser.

## Role of the Learning Management System in security

In order to ensure the security of an LTI platform, the LMS funneling into it via LTI federation must in turn also be secure. An attack leveled at the LMS could in turn become a security breach into any LTI platform it connects with. LMSs must ensure

users are properly authenticated and are who they say they are. The LMS must ensure there is no possibility of [privilege escalation](#).

Of key importance, the LMS must ensure that shared secrets remain secret. It must ensure that it only [digitally signs](#) LTI launch requests that are valid and permissible for the user and context in question, with the correct roles assigned. The LMS must also ensure that it defends against [man in the middle attacks](#), and that [sessions cannot be hijacked](#).

## Security of the connection to the LMS

In order to guarantee that users are using their LMS in the most secure manner possible, the servers hosting them should be using hardened TLS with Strict Transport Security.

If you wish to secure a platform with TLS you will need to:

1. [Purchase a certificate](#) from one of several Certificate Authorities.
  - If you have multiple platforms running on different [subdomains](#), you can [purchase a wildcard certificate](#).
  - If you have multiple primary domains that you need to secure, or you only need to secure a few sub-domains and do not require an unlimited amount, [purchase a multi-domain certificate](#).
2. Set up your web server and platform to use HTTPS with your certificate.
3. [Ensure your web server is set up correctly](#).

MEETS by default records then rejects incoming requests that it can determine are via an LMS connection not using HTTPS.

## Security of the connection to the Web Conference

In leveraging web-conferencing [Application Programming Interfaces](#), MEETS will only communicate over HTTPS with TLS using a hardened [cipher suite](#). Third party servers have their [certificates](#) and [certificate chain](#) validated.

## Web-conferencing credential security

When web-conferencing credentials are entered into MEETS, they are [encrypted](#) using the [Advanced Encryption Standard](#), [leading finalists](#), and stronger more recently developed algorithms. The encryption performed uses two keys, each making use of the largest key size the algorithm in use supports. The first key is randomly generated for each individual client and compiled directly into the

MEETS platform. The second key is randomly generated for each individual user, and is stored in a completely separate [database](#) from the encrypted credentials.

The software unit responsible for the encryption and decryption of the credentials is the same unit used to communicate with the web conferencing service, which is separated from the software used by the rest of the MEETS platform. This unit is the only software component allowed to access the database containing the encrypted credentials, and is limited to using the credentials solely with the web-conferencing APIs. If a user needs to access the web-conferencing service from their web-browser, a limited access or [one-time-use](#) token is generated for them, as this software unit is unable to provide the rest of the MEETS platform with any kind of access to the credentials.

An attacker looking to retrieve credentials would need to steal the primary MEETS platform database, the security unit's database which runs on a completely separate database platform, and the software itself, which must be then decompiled in search of the client key.

The MEETS platform uses separate sets of databases for each client. This ensures that there is no unified target to attack, that any optional security settings declined by one client which lower security will not adversely affect other clients who did not agree to lower security levels, and that any attacker wishing to gain access would have to target each individual client separately.

## **Web-conferencing [privilege separation](#)**

Teachers are given limited access or one-time-use tokens in order to be logged into a conferencing system and launch a web-conferencing session. MEETS provides the option to allow one teacher to substitute for another teacher. Some web-conferencing services may allow that the one logged in for hosting a session is further able to perform other actions with the web-conferencing service beyond session hosting activities.

Therefore, it is strongly advised that for web-conferencing services where applicable, that no teacher account is granted administration privileges. Even if a client only has a single host with their web-conferencing service, several web-conferencing services allow the administration and hosting capabilities be split across two separate users. All supported web-conferencing services that do not offer the aforementioned privilege separation capabilities, thankfully, prevent those logged-in solely for hosting a meeting from performing account-wide activities.

## MEETS Administration Authentication

MEETS uses the strongest level of [HTTP Digest Authentication](#) that browsers currently support. The implementation is hardened to the same extent as the LTI endpoint where applicable, which is described above.

## Other Access

Subsequent accesses to MEETS where a user is known to be authenticated makes use of tokens to prevent [Cross-site request forgeries](#).

Internal communication between different MEETS components makes use of [Authenticated Encryption](#) leveraging [Secure Hash Algorithm 3 \(SHA-3\)](#) finalists along with more recent improvements in the field.

External communication as well as communication between various MEETS components is untrusted until proven otherwise. Input validity and authorization checks are performed at each level as needed. Data is either rejected or sanitized to prevent a variety of breaches and attacks.

Internal databases are only accessed with external variables passed to [parameterized statements](#). Access between other components which leverage programming statements computed at run time makes use of [data escaping](#) of external variables as required.

In short, CirQlive is mindful of the various programming pitfalls and attacks that are commonplace, and strives to remain ahead of the curve in protecting its software, its services, and its users.

## MEETS Environment

MEETS runs in a heavily locked down and limited environment, with the [attack surface](#) limited to what is required for MEETS functionality and its administration and maintenance. Extraneous [network services](#) are not installed, nor are their [ports](#) enabled.

Remote access to the MEETS [servers](#) for administration by CirQlive staff utilizes [multiple layers of protection](#). This includes at least two authentication steps to access any data or perform service level changes. CirQlive staff are not given more access than their role requires.

Countermeasures are employed to prevent unauthorized access, as well as appear invisible to common Internet scans hunting for servers to attack, and foil [brute force](#) and [dictionary](#) attacks.

## Development Practices

### Training

All CirQlive developers are trained in safe design practices, cryptography engineering, and attack countermeasures. Developers are frequently given refresher courses and updated training as security research reveals new insights that require awareness. CirQlive strives to surpass industry standard best-practices and therefore does not limit itself to a single methodology or source of information regarding proper design.

Some of the resources used to train developers are:

- [Bulletproof SSL](#)
- [Code Complete \(2\)](#)
- [Cryptography Engineering](#)
- [NIST Computer Security Resource Center](#)
- [NSA Information Assurance Directorate](#)
- [Open Web Application Security Project](#)
- [Secure Programming Cookbook](#)
- [Secure Programming HOWTO](#)

Developers are trained to understand the information contained within these resources, to apply their material appropriately, and to weigh the pros and cons of various strategies to arrive at correct designs when conflicts or concerns arise. Additional resources are provided to developers as needed specific to the field they are working in.

### MEETS in-house use

CirQlive uses MEETS to conduct in-house training. This ensures that developers are familiar with it, and that the product is well tested with real world usage. It also consistently provides CirQlive with first-hand insight into improving the product.

### Design and Implementation

As part of designing our systems, CirQlive gathers requirements and performs [threat modeling](#) and [vulnerability analysis](#) upon each component individually and as a whole. Designs which are vulnerable are disregarded, restarting the process of

creating a secure design. Upon completion, modification, or integration of components, in-house [white-box penetration testing](#) is performed using a variety of manual and automatic processes in order to validate security requirements. Newly written code and modifications are reviewed and audited by in-house security personnel using [static analysis](#) and human-based analysis to ensure there are no security vulnerabilities. CirQlive does not deploy components or products with known security defects.

## Security Research

CirQlive's security team stays up-to-date with the latest security research, and, in addition to regularly monitoring deployed platforms, performs updated testing and threat assessments any time a new kind of attack is discovered which may be relevant. Components vulnerable to any such newly discovered attacks are typically updated in a timely fashion. Usually, such vulnerabilities require only a minor configuration change to disallow older algorithms or methodologies. In the rare case a newly discovered kind of attack requires restructuring, CirQlive immediately dedicates a team to redesign the affected components and deploy them using our secure development methodology.

Some of the sources CirQlive utilizes for knowledge about new attacks are:

- [Bulletproof TLS Newsletter](#)
- [Debian Security Advisories](#)
- [LWN.net security alerts](#)
- [OpenSSL Vulnerabilities](#)
- [Qualys Security Labs](#)
- [Qualys Web Application Security](#)
- [US-CERT Alerts](#)
- [US-CERT Bulletins](#)

The CirQlive security team also monitors [blogs](#) from various security researchers, and participates in security forums and [IRC](#) channels.

## Technology

MEETS uses software developed by the OpenSSL Project for use in the [OpenSSL Toolkit](#).

## Certifications

The majority of US [data-centers](#) used by CirQlive have obtained [SSAE 16 SOC](#) compliance. If necessary, you can request that CirQlive only place your MEETS account within a US data-center that has obtained SOC2 compliance.

Although MEETS does not perform Credit Card processing, nor collect personal identification (ID numbers, Social Security, etc...), the data-centers used by CirQlive are [PCI DSS](#) (v3.2) compliant. Although MEETS does not directly offer health services, these data-centers are also [HIPAA](#) compliant. Further, the TLS settings described at the beginning of this document meet or exceed the baseline requirements set forth by PCI DSS and HIPAA in these areas.

MEETS regularly receives an A+ security rating from [SSL Labs](#), [Mozilla Observatory](#), [CryptCheck](#) and others. According to [SSL Pulse](#), as of May 2015, only 1.4% of the most popular sites surveyed with TLS deployed achieve an A+ rating. When including all sites within the top one million, including those without TLS deployed, this number drops to less than 0.2%.