**Bosch Security Systems** | Video Systems

# Network Authentication - 802.1X

*Secure the Edge of the Network - Technical White Paper*

BOSCH
Invented for life

**14 September 2015**

## Secure the edge of the network

Security devices are mostly located at the physical edge of the network. Especially detection devices, such as cameras, are installed in places that are accessible by the public. As these devices are connected to the network, this also increases the risk of unwanted access to the network: people could try to disconnect the security device and connect their own equipment to try to gain access to the network, or attach pass-through equipment to try a so-called a man-in-the-middle attack.
There are several ways of mitigating such attempts:

- Ensure the device meets the requirements related to physical strength and cabling management: Bosch devices that have an IP66 or IP67 rating have this network connection point inside their housing. This means they need to be physically disassembled before the network connection point can be accessed. This can be further secured by using tamper-proof screws.
- Authenticate the device to the network before allowing it to access the network's resources: there are several ways to ensure that only authenticated devices can access the network. Bosch devices support authentication based on user-name and password (802.1x). In addition to 802.1X EAP-TLS can be used, which secures the whole authentication process.

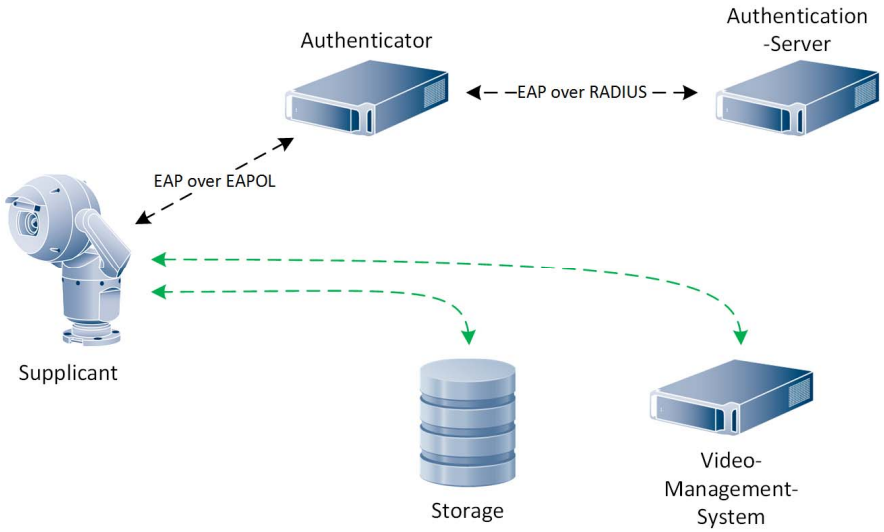



*Figure 1 - EAP (Extensible Authentication Protocol) data is first encapsulated in EAPOL frames between the Supplicant and Authenticator, then re-encapsulated between the Authenticator and the Authentication server using RADIUS or Diameter.*

### IEEE 802.1X

IEEE 802.1x [1] is a standard published by the Institute of Electrical and Electronics Engineers Standards Association. This organization within the IEEE develops global standards in a broad range of industries, including: power and energy, biomedical and health care, information technology, telecommunication, transportation, nanotechnology, information assurance and many more. This particular standard is part of the IEEE 802.1 group of networking protocols and provides an authentication mechanism to prevent unauthorized devices to access network resources.

This protocol involves three kinds of main elements:

- The element that wants to be able to access the network resources is called the supplicant, for example a video surveillance camera.
- The element that verifies if the supplicant may access the network resources is called the authenticator. Mostly this is a manageable switch, router or wireless access point.
- The element that actually steers the authentication process is called the authentication server. The authentication server contains the information which is used to decide if a supplicant may or may not access the network resources. Typically this is a server which supports the RADIUS protocol, which is a networking protocol that provides centralized authentication, authorization and accounting. The RADIUS protocol is part of the Internet Engineering Task Force (IETF) standards.

### Extensible Authentication Protocol

The Extensible Authentication Protocol [2] is an authentication framework which supports multiple authentication methods. EAP typically runs directly over data link layers such as Point-to-Point Protocol (PPP) or IEEE 802, without requiring IP. EAP provides its own support for duplicate elimination and retransmission, but is reliant on lower layer ordering guarantees.

A typical EAP authentication procedure using RADIUS [3] consists of four steps:

1. Initialization:
   After the authenticator detects that a device is connected to its port, this port is set to the "unauthorized" state and will only allow 802.1X traffic. Other traffic, such as UDP or TCP is not allowed and dropped.
2. Initiation:
   The authenticator will request the identity of the supplicant. When the authenticator receives this information it will forward it to the authentication server by means of the RADIUS protocol.
3. Negotiation:
   The authentication server verifies the supplicant identity and sends a challenge back to the supplicant via the authenticator. This challenge also contains the authentication method, which could be based on a user-name and password.
4. Authentication:
   The authentication server and supplicant agree on an authentication method and the supplicant will

respond with the appropriate method by providing its configured credentials. If authentication is successful, the authenticator allows the supplicant access to the defined network resources.

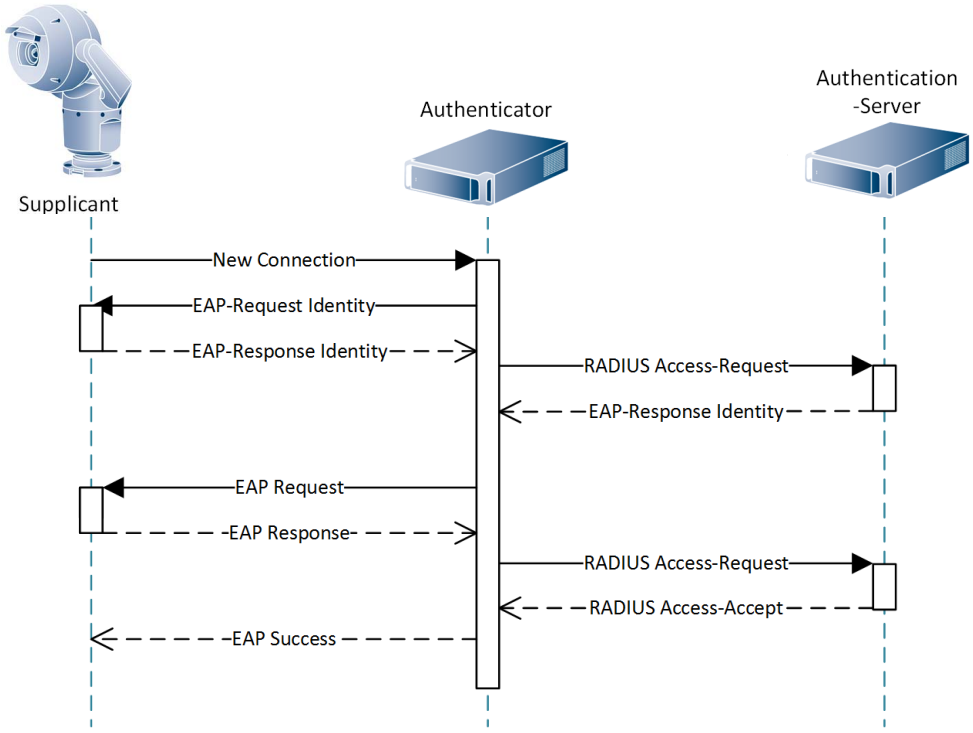



*Figure - 802.1X Authentication Sequence Diagram*

802.1X itself does not provide a secure communication between the supplicant and authentication server. As a result the user-name and password could be "sniffed" from the network. To ensure a secure communication 802.1X can use EAP-TLS.

### Extensible Authentication Protocol - Transport Layer Security

The Extensible Authentication Protocol (EAP), provides support for multiple authentication methods. Transport Layer Security (TLS) provides for mutual authentication, integrity-protected cipher suite negotiation, and key exchange between two endpoints. EAP-TLS [4] includes support for certificate-based mutual authentication and key derivation. In other words, EAP-TLS encapsulates the process in which both the server and client send each other a certificate.

### Certificates

Digital certificates are used to verify that a public key belongs to a specific user or device, in other words, to verify that a user or device is actually telling the truth about its identity. These certificates are generated by a Certificate Authority (CA) based on specific details of the user or device. This Certificate Authority needs to be a trusted entity within an entire infrastructure and ensures that the certificates that are used in the infrastructure can be verified. A compromised Certificate Authority cannot be trusted any more, and can therefore also not verify the identity of a user or device.

The management of certificates, assertion, extension and revocation, is typically handled within a Public Key Infrastructure (PKI) [5].

### EAP-TLS client certificate

The EAP-TLS client certificate binds the client's identity to a public key. This public key (as part of the certificate) is send to the server and used to encrypt the communication between the client and server.

The requirement for a client-side certificate is what gives EAP-TLS its authentication strength. With a client-side certificate, a compromised password is not enough to break into EAP-TLS enabled systems because the intruder still needs to have the client-side certificate. The highest security available is when the "private keys" of client-side certificate are housed in smart cards, or in Trusted Platform Modules in devices like Bosch's security cameras. This is because there is no way to steal a client-side certificate's corresponding private key from a smart card or TPM without stealing the card itself – or the security camera. In the latter case, still no one could make use of it other than trying to connect the device to the network it is dedicated for.

### EAP-TLS trusted certificate

When a client is presented with a server's certificate, the client tries to match the server's Certificate Authority (which is part of the certificate) against the client's list of trusted Certificate Authorities. If the issuing Certificate Authority is trusted, the client will verify that the certificate is authentic and has not been tampered with. Finally, the client will accept the certificate as proof of identity of the server.

## Certificates in Bosch cameras

All Bosch cameras (FW 6.10 or newer) use a certificate store, which can be found in the ***Service*** section of the camera configuration. Both the EAP-TLS client certificate and the EAP-TLS trusted certificate need to be added to the store by using the ***Add*** button in the ***File list*** section. After the upload is completed the certificates can be selected in the ***Usage list***. To activate the use of the certificates the camera must be rebooted, which happens automatically after pressing the ***Set*** button, and 802.1x must be activated in ***Network->Advanced*** with credentials entered.

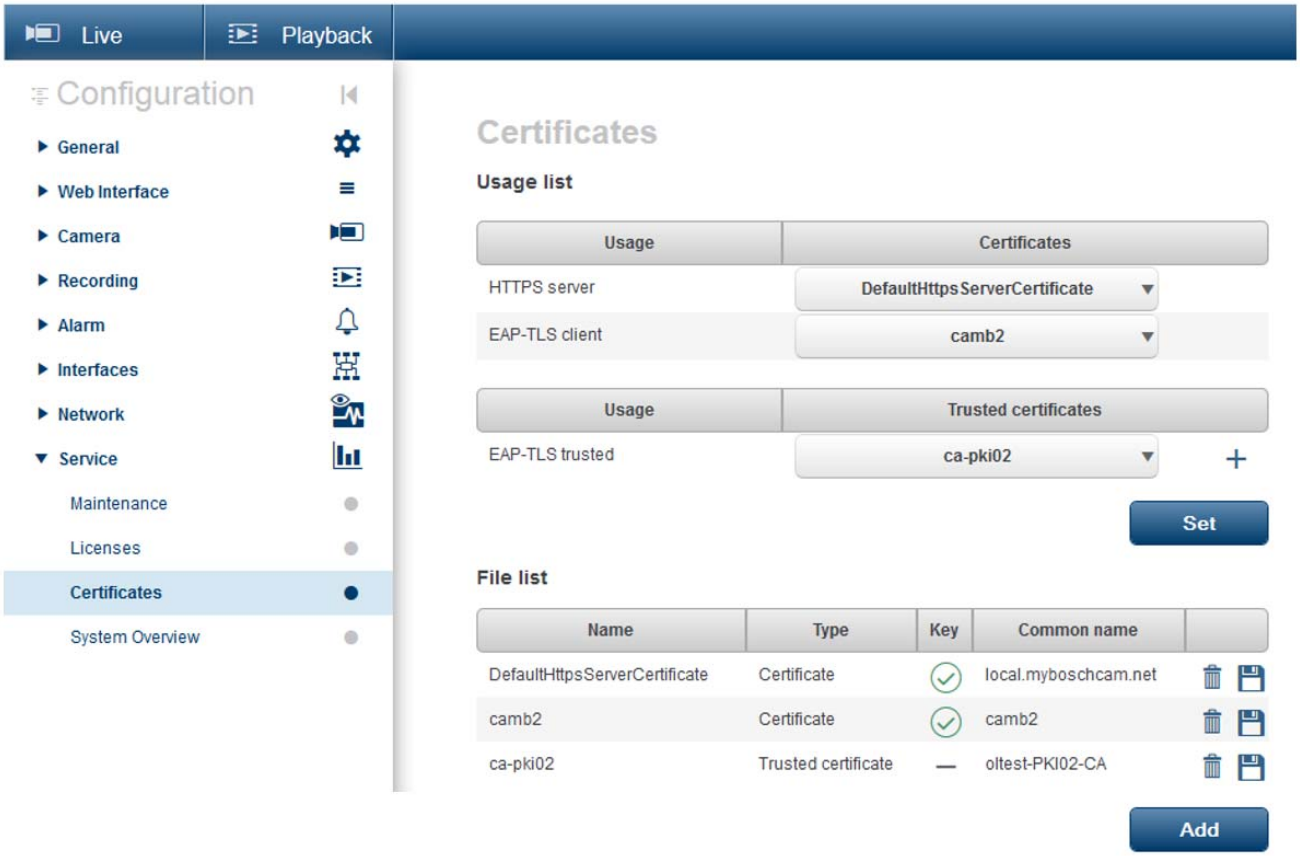



*Figure3 – Example: EAP/TLS certificates stored in a Bosch camera (FW6.11)*

### Secured in a safe

The certificates are stored in a chip like being used on SmartCards, also called a “Trusted Platform Module”, or short TPM. This chip acts like a safe for critical data, protecting certificates, keys, licenses, etc. against unauthorized access even when the camera is broken up.

Certificates are accepted in **.pem* format. They may be uploaded as one combined file, or split into certificate and key parts and uploaded as separate files.

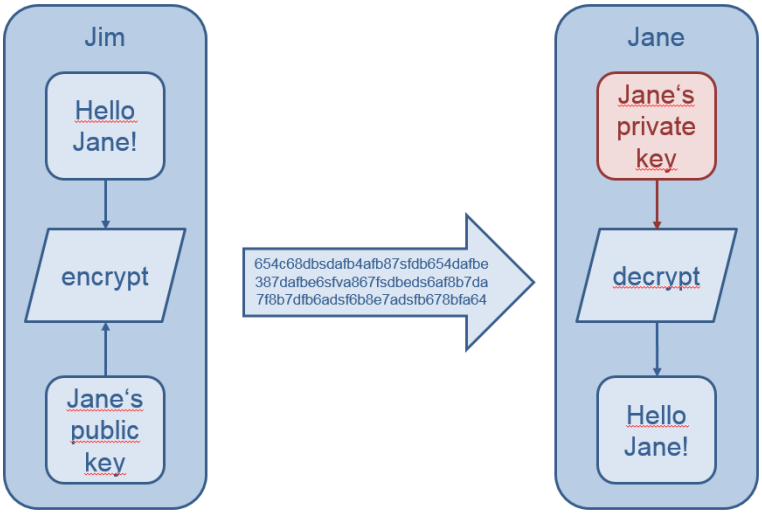
# Appendix

This appendix contains a description of how public and private keys are used to encrypt network communication.

## Public key encryption

Public key sizes vary from 512 bits to 2048 bits. Because of the large size, public keys are extremely slow and generally not feasible for bulk data encryption. However, public keys are widely used for user and device authentication.

Examples include: Digital Certificates, RSA




Hash functions are used to provide integrity services. A hashing algorithm has to be one way; it's not possible to determine the original data from the hash value. The hash function also has to be collision resistant. A collision occurs when two different inputs give the same output. Above all, it should not be possible to predict a different input value that will give the same output.

The most secure hash function that is widely used at present is the SHA-1. SHA-1 is preferred over MD5, although MD5 is still widely supported. Fueled by increased computational power available, SHA-256 takes over due to its improved security provision. Bosch cameras allow selection of one of these three for e.g. video authentication. These functions produce a fixed-length output, which is useful when working with IP packets because the overhead of transmitting the hash value is predictable.

Source authentication is performed using the Hashed Method Authentication Code (HMAC).

1. The sender appends a secret pre-shared key to the data
2. The sender performs a hash function over the data
3. Receiver must append same key value to the data before performing the hash function

The key itself is never transmitted along with the data.

### Diffie Hellman Key Exchange

The Diffie Hellman (DH) algorithm is a method whereby two parties can agree upon a secret key that is known only to them. The strength of the technique is that it allows the participants to create the secret value over an unsecured medium without exchanging the secret value itself. It is also impossible to reverse-generate the secret if it is somehow intercepted.

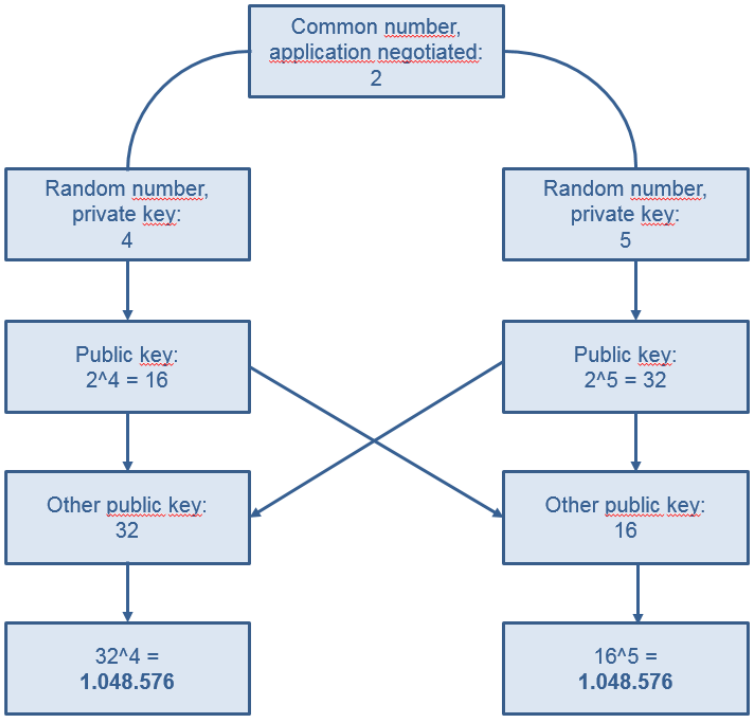
DH groups consist of 5 groups of very large prime numbers (and a generator) used as the modulus for the DH algorithm, Juniper equipment supports group 1, 2 and 5:

- Group 1 uses a 768 bit prime
- Group 2 uses a 1024 bit prime
- Group 5 uses a 1535 bit prime

The larger the prime number, the stronger the key, and the more computationally intensive the calculation.

Using the same DH group, each device creates a unique public/private key pair. These keys are mathematically related using the DH algorithm.

- The public key values are exchanged across the network
- Each peer then runs its local private key and the received public key value through the DH algorithm to compute a common session key
- The session key itself is never passed across the network




Just to give an impression, this is a 768 bit, or 232 digit number as being used for group 1 if a prime:

*3246872364872864762746854468744382736878321218721387468765837246823762876834724368763874218741286475713491387491283471298347637832153847938541328749837581327497438217489374981274897348571981374981379487918374193847193547198749187341*

## Authors

Mario Verhaeg (ST/SBD-EU)

Konrad Simon (ST-VS/MKP1)

## References

1 802.1x, IEEE standard for port-based Network Access Control
http://www.ieee802.org/1/pages/802.1x-2004.html

2 RFC 3748, Extensible Authentication Protocol (EAP),
https://tools.ietf.org/html/rfc3748

3 RFC 3580, IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines
https://tools.ietf.org/html/rfc3580

4 RFC 5216, The EAP-TLS Authentication Protocol,
http://www.ietf.org/rfc/rfc5216.txt

5 RFC 3280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
https://www.ietf.org/rfc/rfc3280.txt

Bosch Sicherheitssysteme GmbH
Robert-Bosch-Ring 5
85630 Grasbrunn
Germany
www.boschsecurity.com