Robust and Scalable Security Monitoring and Compliance Management for Dynamic Energy Delivery Systems

Key Distribution for the Internet of Energy

R. Gentz, A. Scaglione, S. Peisert, C. McParland, T. Edgar, E. Choi, M. Engels

GOALS

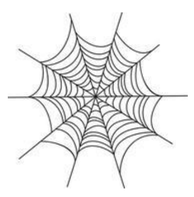
- Develop a secure and resilient **key distribution** architecture that supports **millions** of devices.
- Centrally managed.
  - Group policy enforcement and auditing.
  - Must be able to revoke keys.
- Must be disruption-tolerant.
  - Must continue to operate for a given time, even if network communications are cut off.
- Automated key management for low-power devices (computation/memory).
  - Resolve entropy problem.
- Allow authorized third-party access.

CURRENT APPROACHES

- Public key infrastructure

  - Central certificate authorities.
  - Authorize with a chain of trust.
    - Third-party authorization is complicated.
  - Revocation of certificates:
    - CRL (static lists) will get large quickly; impractical on low-power devices.
    - OCSP (online check): What happens in case of lost connectivity?
      - Accept everything or reject everything?
        - » Neither is acceptable.

- Pretty Good Privacy (PGP):



  - Web of trust.
    - (I trust you, you trust him → I trust him).
  - Completely distributed.
  - No central control or enforcement.
  - Key revocation requests need to be sent to the whole network.
    - Impractical.
  - Third-party authorization not accounted for.

OUR APPROACH

- **Combine** ideas from enterprise key management, identification, and authorization protocols.

- **Kerberos**
  - *Cached* central authorizations.
  - Central policy enforcement.
  - Disruption-tolerant.
  - Keys expire with a short lifetime (no need to revoke).

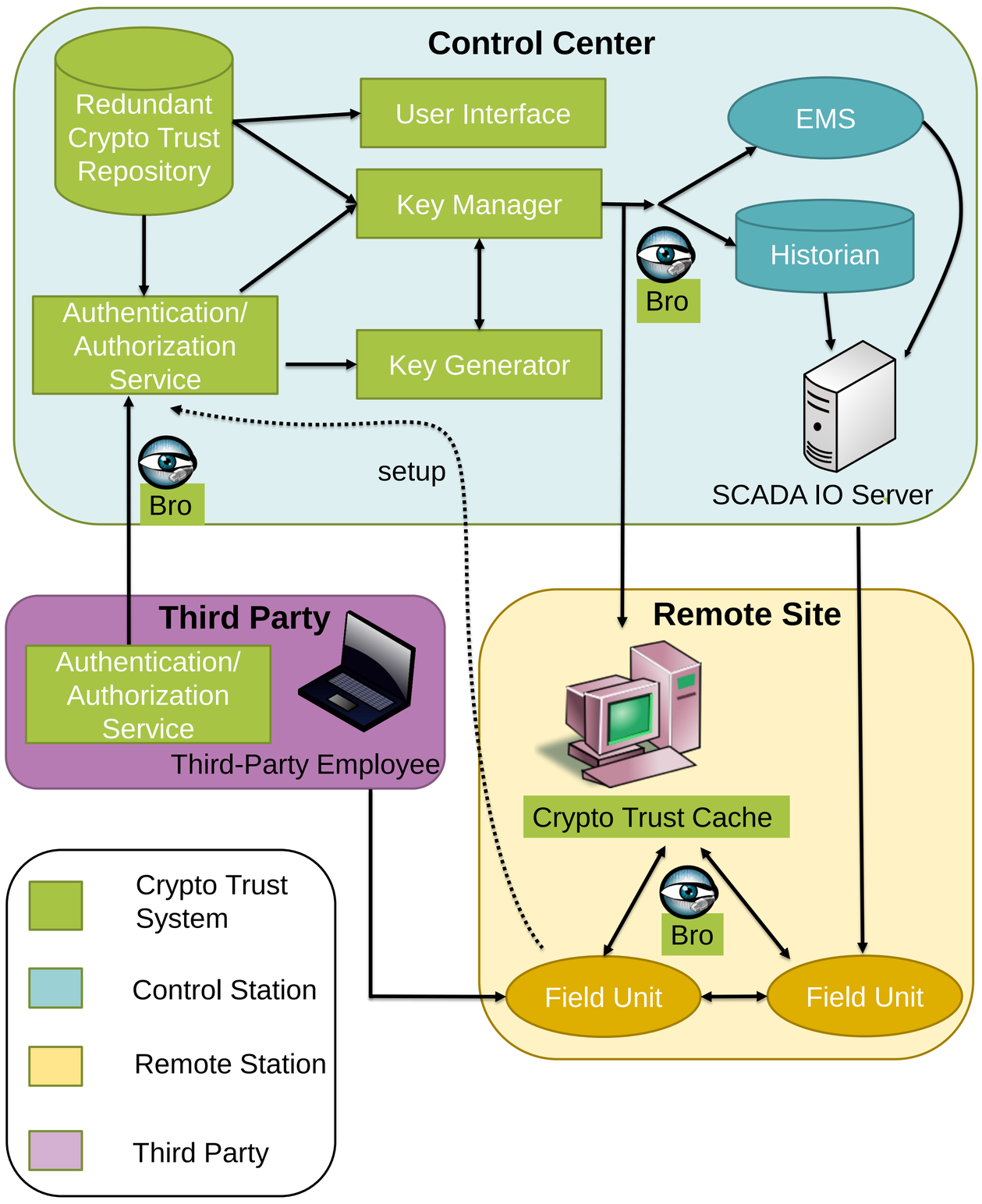
- **802.1ar**
  - Device identity and authentication.

- **802.1x**
  - Tunneled authorization of not yet authenticated devices.

- **KMIP** (Key Management Interoperability Protocol)
  - Automated key registration/update.
  - Group keying.
  - Support for legacy devices.

- **Bro**
  - Monitor the network.
  - Key management and authentication systems are high-value targets.
  - Add cheap remote sensors, like BeagleBoard or Raspberry Pi, in remote site to monitor correct key management behavior.

OUR APPROACH



WHAT HAPPENS…

- … if communication to the remote site breaks?
  - Crypto Cache provides field devices with key material until its timer runs out. (This provides enough time to fix the fault.)
- … if a field device is compromised?
  - The field device will no longer receive key updates. Because of the short lifetime of keys, other devices will no longer trust the compromised device.
- … internally when a new field device is added?
  - The device is provided with a secure tunnel to the authentication service, while being excluded from the rest of the network.
  - After authentication is complete, the device generates a public key pair, or receives one if it is not capable of key generation.
  - The key is signed by the central authority for a limited time.
  - The field device can now communicate with other devices.
- … if the signature of the authority times out?
  - The device can receive an extension from the trust cache or a new signature from the central service.
- … if I want to add a legacy device?
  - KMIP allows integration of field devices that do not support the proposed architecture.

PROJECT OUTCOMES

- Prototype key management system.
- Bro-based defense architecture.
- Test scenarios and models.
- Performance validation.
- Comparison with other key management systems (IEC052351, NISTIR 7628).
- Research platform with test applications for future security research.

FUNDING SUPPORT PROVIDED BY THE DEPARTMENT OF ENERGY, OFFICE OF ELECTRICITY DELIVERY & ENERGY RELIABILITY