Bootstrapping Trust in Networked Measurement Systems with Secure Sensors

Kristján Valur Jónsson and Ýmir Vigfússon

Reykjavik University, School of Computer Science
Menntabraut 1, 101 Reykjavik, Iceland
{kristjanvj,ymir}@ru.is

February 8, 2012

Reykjavik University
http://www.ru.is
School of Computer Science

Reykjavik University
http://www.ru.is
School of Computer Science

Reykjavik University
http://www.ru.is
School of Computer Science

Correct and reliable sensing in networked systems

- Sensing *accuracy* is important
- Minimize interference and other sources of error
- Sensor information: **power to make intelligent decisions**

Correct and reliable sensing in networked systems

- Sensing *accuracy* is important
- Minimize interference and other sources of error
- Sensor information: **power to make intelligent decisions**

- Cooperative data contributors
- *common goal*: production of "good" measurements

Correct and reliable sensing in networked systems

- Sensing *accuracy* is important
- Minimize interference and other sources of error
- Sensor information: **power to make intelligent decisions**

- Cooperative data contributors
- *common goal*: production of "good" measurements
- What if this is not the case?

Correct and reliable sensing in networked systems

- Sensing *accuracy* is important
- Minimize interference and other sources of error
- Sensor information: **power to make intelligent decisions**

- Cooperative data contributors
- *common goal*: production of "good" measurements
- What if this is not the case?

- **Dishonest world**: Some parties want to *influence* measurements.

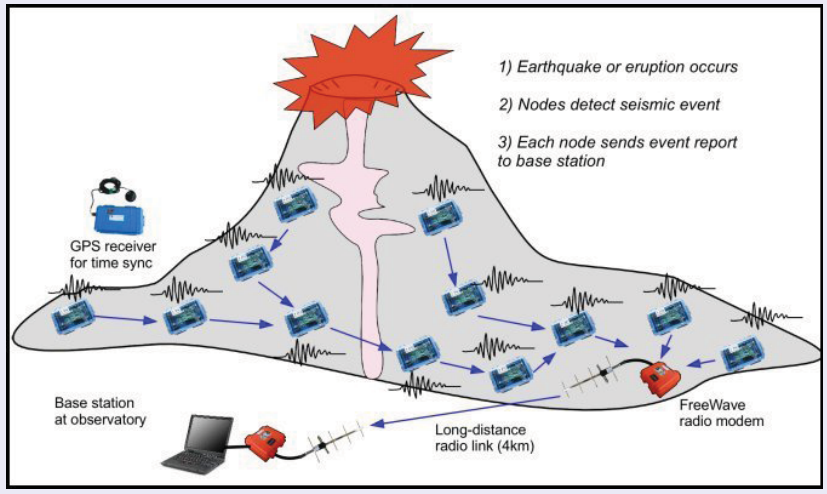
Correct and reliable sensing in networked systems

- Sensing *accuracy* is important
- Minimize interference and other sources of error
- Sensor information: **power to make intelligent decisions**

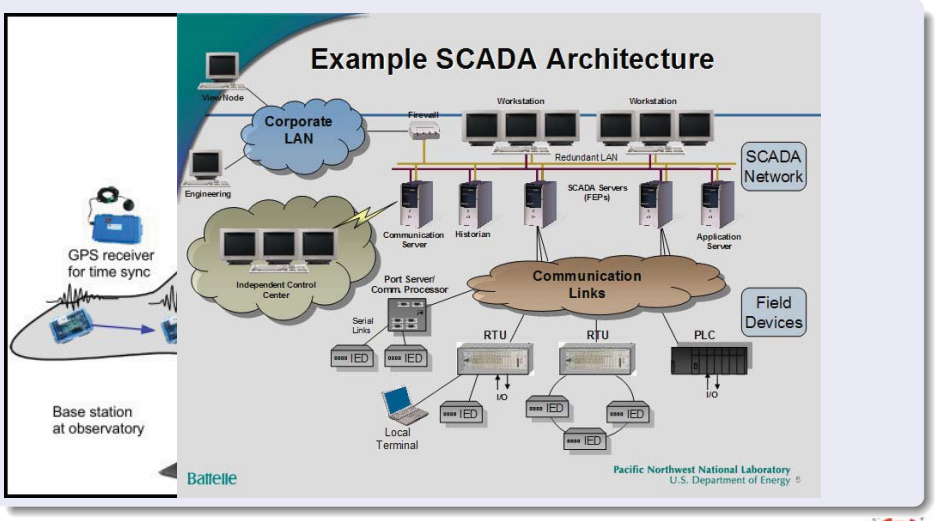
- Cooperative data contributors
- *common goal*: production of "good" measurements
- What if this is not the case?

- **Dishonest world**: Some parties want to *influence* measurements.
- How do we approach this problem?

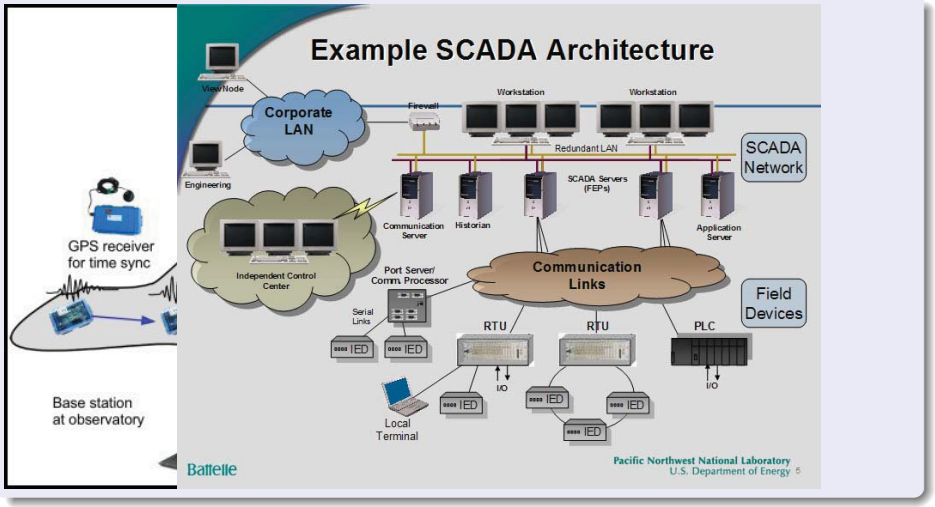
Sensing in networked systems



Sensing in networked systems



Sensing in networked systems



Observation node: Any networked node which carries a sensor

Security objectives and adversarial model

Objectives

- *Primary:* Sensor data **integrity**.
  - The *consumer* of the data can be confident in the *integrity* of delivered data, *even if some participants are corrupt*.

Security objectives and adversarial model

Objectives

- *Primary:* Sensor data **integrity**.
  - The *consumer* of the data can be confident in the *integrity* of delivered data, *even if some participants are corrupt*.
- *Secondary:* **confidentiality**

Security objectives and adversarial model

Objectives

- *Primary:* Sensor data **integrity**.
  - The *consumer* of the data can be confident in the *integrity* of delivered data, *even if some participants are corrupt*.
- *Secondary:* **confidentiality**

Adversarial model

- **Adversary**: Can corrupt *observation nodes*
- **Active insiders**: Can observe and rewrite data
- **Stealthy operation**: Corrupt nodes try to evade detection

Security objectives and adversarial model

Objectives

- *Primary:* Sensor data **integrity**.
  - The *consumer* of the data can be confident in the *integrity* of delivered data, *even if some participants are corrupt*.
- *Secondary:* **confidentiality**

Adversarial model

- **Adversary**: Can corrupt *observation nodes*
- **Active insiders**: Can observe and rewrite data
- **Stealthy operation**: Corrupt nodes try to evade detection
- Disregard: privacy, availability and process-of-measurement attacks

Establishing system trust

Establishing system trust

Approaches

1. *Ostrich* – blind trust

Establishing system trust

Approaches

1. *Ostrich* – blind trust
2. Harden observation nodes

Establishing system trust

Approaches

1. *Ostrich* – blind trust
2. Harden observation nodes
3. **Harden the essential functionality**

Establishing system trust

Approaches

1. *Ostrich* – blind trust
2. Harden observation nodes
3. **Harden the essential functionality**

Essential functionality

- *Primary product:* trustworthy data.
- *Essential functionality:* Integrity-preserving sensing

Establishing system trust

Approaches

1. *Ostrich* – blind trust
2. Harden observation nodes
3. **Harden the essential functionality**

Essential functionality

- *Primary product:* trustworthy data.
- *Essential functionality:* Integrity-preserving sensing

Trusted sensor: Tool for secure sensing

Trusted sensor concept

Proposal

Model the trusted sensor on the *reference monitor* construct from classical trusted systems theory

Trusted sensor concept

Proposal

Model the trusted sensor on the *reference monitor* construct from classical trusted systems theory

Trusted sensor

- Functionality rigorosly *specified*

Trusted sensor concept

Proposal

Model the trusted sensor on the *reference monitor* construct from classical trusted systems theory

Trusted sensor

- Functionality rigorosly *specified*
- *Verifiable* – preferably *formally verifiable*

Trusted sensor concept

Proposal

Model the trusted sensor on the *reference monitor* construct from classical trusted systems theory

Trusted sensor

- Functionality rigorosly *specified*
- *Verifiable* – preferably *formally verifiable*
  - *Small* in terms of code, circuits and functionality

Trusted sensor concept

Proposal

Model the trusted sensor on the *reference monitor* construct from classical trusted systems theory

Trusted sensor

- Functionality rigorously *specified*
- *Verifiable* – preferably *formally verifiable*
  - *Small* in terms of code, circuits and functionality
  - *Unique identity* – (Public ID, Private cryptographic key K_{As})

Trusted sensor concept

Proposal

Model the trusted sensor on the *reference monitor* construct from classical trusted systems theory

Trusted sensor

- Functionality rigorosly *specified*
- *Verifiable* – preferably *formally verifiable*
  - *Small* in terms of code, circuits and functionality
  - *Unique identity* – (Public ID, Private cryptographic key K_{As})
- *Tamper-resistant*

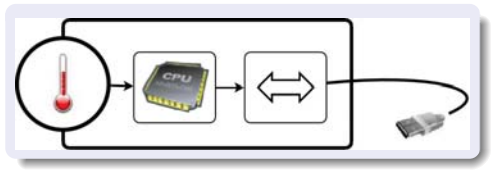
Trusted sensor concept

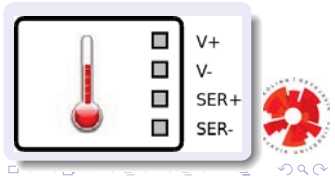
Proposal

Model the trusted sensor on the *reference monitor* construct from classical trusted systems theory

Trusted sensor

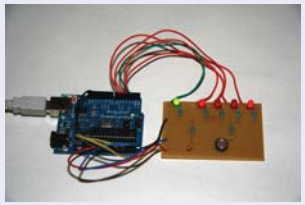
- Functionality rigorosly *specified*
- *Verifiable* – preferably *formally verifiable*
  - *Small* in terms of code, circuits and functionality
  - *Unique identity* – (Public ID, Private cryptographic key K_{As})
- *Tamper-resistant*





Sensor prototype implementation (2010)

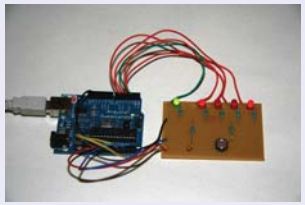
Sensor



- Simple temperature and luminosity sensor
- Arduino Duemilanovae
- ATmega328 CPU

Sensor prototype implementation (2010)

Sensor



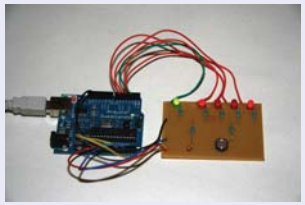
- Simple temperature and luminosity sensor
- Arduino Duemilanovae
- ATmega328 CPU

Cryptographic primitives

1. *Encryption*: **AES-128** (Rijndael) in *CBC-mode*
2. *Message authentication*: **CMAC** (AES128-CMAC)

Sensor prototype implementation (2010)

Sensor



- Simple temperature and luminosity sensor
- Arduino Duemilanovae
- ATmega328 CPU

Cryptographic primitives

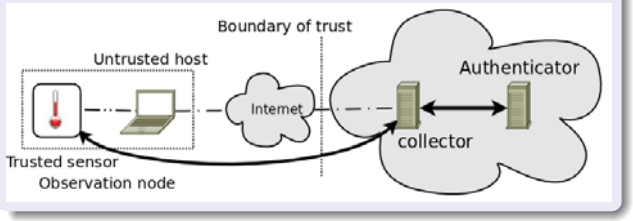
1. *Encryption*: **AES-128** (Rijndael) in *CBC-mode*
2. *Message authentication*: **CMAC** (AES128-CMAC)

Prototype implementation – open source

- 1400 lines of C++ on ATmega328
- **TSense:** `http://code.google.com/p/tsense`
- **Crypto library:** `https://github.com/kristjanvj/ACrypto`

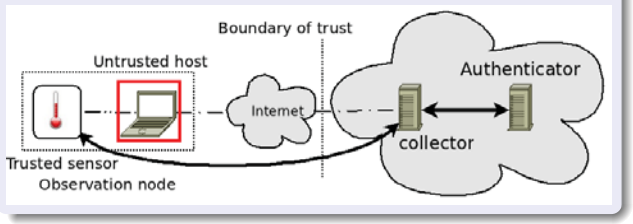
Prototype system

System diagram



Prototype system

System diagram

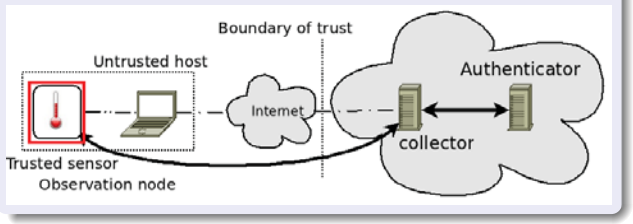


Players

- Vulnerable observation node

Prototype system

System diagram

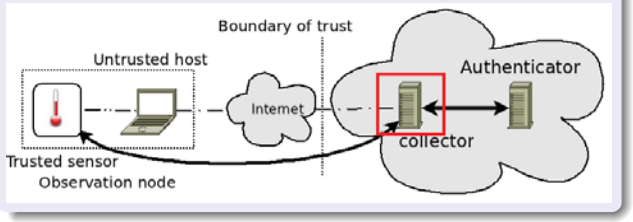


Players

- Vulnerable observation nodes
- Trusted sensors

Prototype system

System diagram

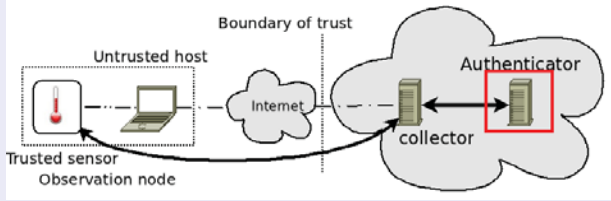


Players

- Vulnerable observation nodes
- Trusted sensors
- Collector

Prototype system

System diagram

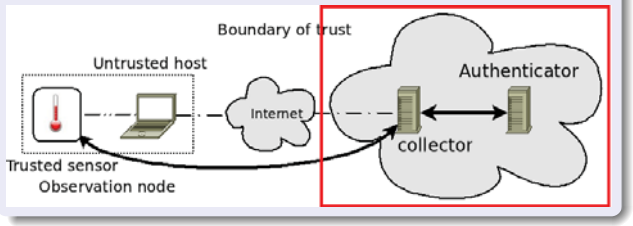


Players

- Vulnerable observation nodes
- Trusted sensors
- Collector
- Authenticator

Prototype system

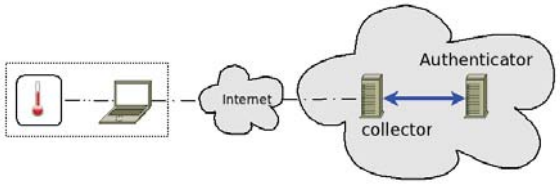
System diagram



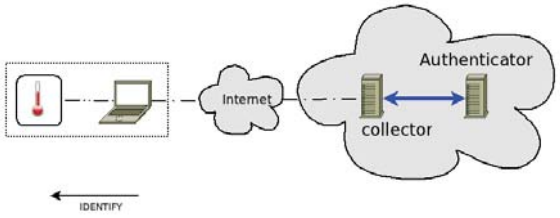
Players

- Vulnerable observation nodes
- Trusted sensors
- Collector
- Authenticator

Protocols I: Authentication

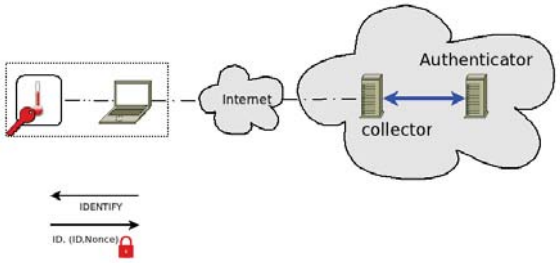


Protocols I: Authentication



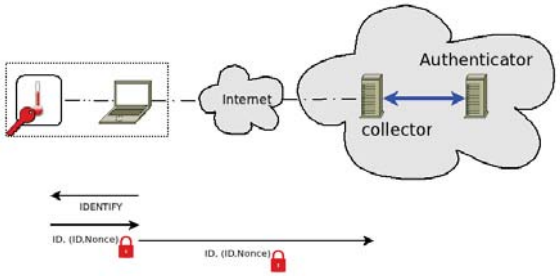
$$
\begin{aligned}
s \leftarrow S: &\quad \langle \text{IDENTIFY} \rangle \\
s \rightarrow S \rightarrow C \Rightarrow A: &\quad \langle \text{AUTH}, s, IV, \mathcal{ET}_{As}(s, N_s) \rangle \\
C \Leftarrow A: &\quad \langle \text{AUTH}, s, IV, K_{Cs}, \mathcal{ET}_{As}(N_s, K_{Cs}) \rangle \\
s \leftarrow S \leftarrow C: &\quad \langle \text{AUTH}, s, IV, \mathcal{ET}_{As}(N_s, K_{Cs}) \rangle
\end{aligned}
$$

Protocols I: Authentication



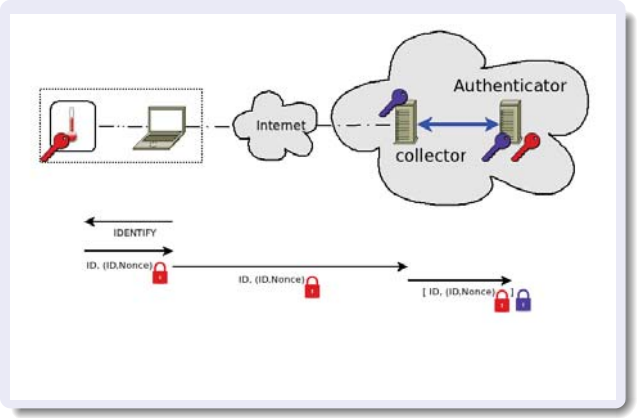
$$
\begin{aligned}
s \leftarrow S: &\quad \langle \text{IDENTIFY} \rangle \\
s \rightarrow S \rightarrow C \Rightarrow A: &\quad \langle \text{AUTH}, s, IV, \mathcal{ET}_{As}(s, N_s) \rangle \\
C \Leftarrow A: &\quad \langle \text{AUTH}, s, IV, K_{Cs}, \mathcal{ET}_{As}(N_s, K_{Cs}) \rangle \\
s \leftarrow S \leftarrow C: &\quad \langle \text{AUTH}, s, IV, \mathcal{ET}_{As}(N_s, K_{Cs}) \rangle
\end{aligned}
$$

Protocols I: Authentication



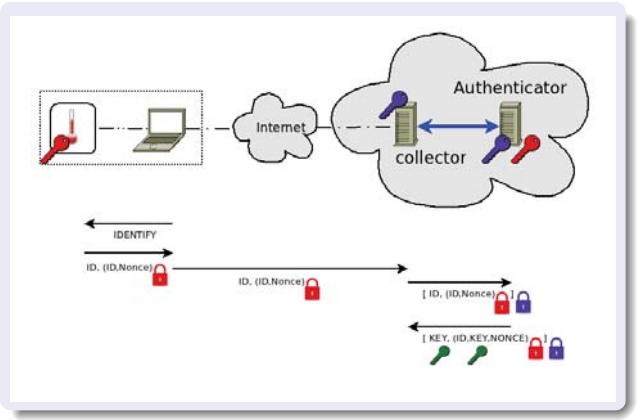
$$
\begin{aligned}
s \leftarrow S: &\quad \langle \text{IDENTIFY} \rangle \\
s \rightarrow S \rightarrow C \Rightarrow A: &\quad \langle \text{AUTH}, s, IV, \mathcal{ET}_{As}(s, N_s) \rangle \\
C \Leftarrow A: &\quad \langle \text{AUTH}, s, IV, K_{Cs}, \mathcal{ET}_{As}(N_s, K_{Cs}) \rangle \\
s \leftarrow S \leftarrow C: &\quad \langle \text{AUTH}, s, IV, \mathcal{ET}_{As}(N_s, K_{Cs}) \rangle
\end{aligned}
$$

Protocols I: Authentication



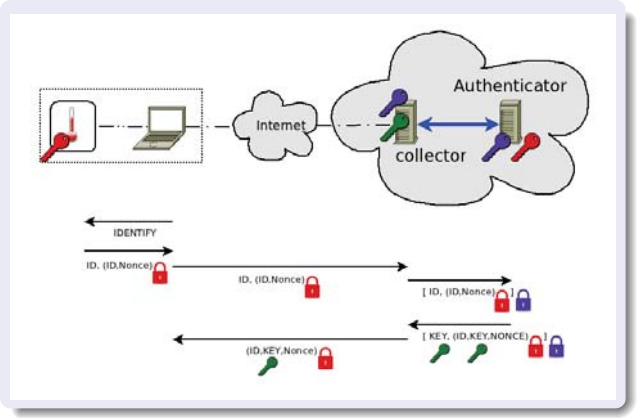
$$
\begin{aligned}
s \leftarrow S: &\quad \langle \text{IDENTIFY} \rangle \\
s \rightarrow S \rightarrow C \Rightarrow A: &\quad \langle \text{AUTH}, s, IV, \mathcal{ET}_{As}(s, N_s) \rangle \\
C \Leftarrow A: &\quad \langle \text{AUTH}, s, IV, K_{Cs}, \mathcal{ET}_{As}(N_s, K_{Cs}) \rangle \\
s \leftarrow S \leftarrow C: &\quad \langle \text{AUTH}, s, IV, \mathcal{ET}_{As}(N_s, K_{Cs}) \rangle
\end{aligned}
$$

Protocols I: Authentication



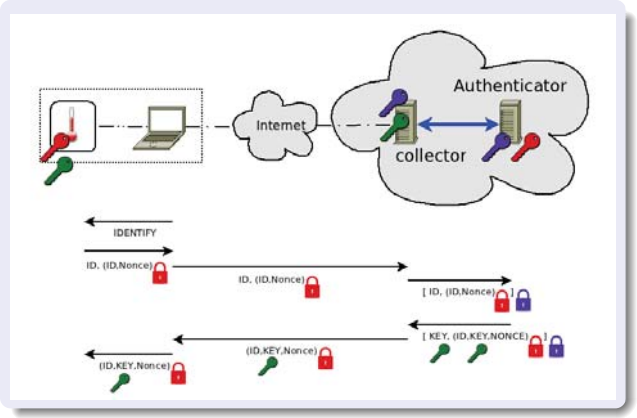
$$
\begin{aligned}
s \leftarrow S: &\quad \langle \text{IDENTIFY} \rangle \\
s \rightarrow S \rightarrow C \Rightarrow A: &\quad \langle \text{AUTH}, s, IV, \mathcal{ET}_{As}(s, N_s) \rangle \\
C \Leftarrow A: &\quad \langle \text{AUTH}, s, IV, K_{Cs}, \mathcal{ET}_{As}(N_s, K_{Cs}) \rangle \\
s \leftarrow S \leftarrow C: &\quad \langle \text{AUTH}, s, IV, \mathcal{ET}_{As}(N_s, K_{Cs}) \rangle
\end{aligned}
$$

Protocols I: Authentication



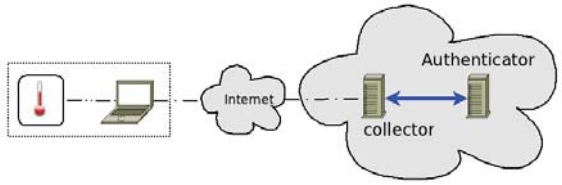
$$
\begin{aligned}
s \leftarrow S: &\quad \langle \text{IDENTIFY} \rangle \\
s \rightarrow S \rightarrow C \Rightarrow A: &\quad \langle \text{AUTH}, s, IV, \mathcal{ET}_{As}(s, N_s) \rangle \\
C \Leftarrow A: &\quad \langle \text{AUTH}, s, IV, K_{Cs}, \mathcal{ET}_{As}(N_s, K_{Cs}) \rangle \\
s \leftarrow S \leftarrow C: &\quad \langle \text{AUTH}, s, IV, \mathcal{ET}_{As}(N_s, K_{Cs}) \rangle
\end{aligned}
$$

Protocols I: Authentication

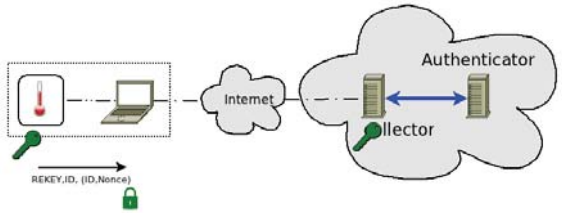


$$
\begin{aligned}
s \leftarrow S: &\quad \langle \text{IDENTIFY} \rangle \\
s \rightarrow S \rightarrow C \Rightarrow A: &\quad \langle \text{AUTH}, s, IV, \mathcal{ET}_{As}(s, N_s) \rangle \\
C \Leftarrow A: &\quad \langle \text{AUTH}, s, IV, K_{Cs}, \mathcal{ET}_{As}(N_s, K_{Cs}) \rangle \\
s \leftarrow S \leftarrow C: &\quad \langle \text{AUTH}, s, IV, \mathcal{ET}_{As}(N_s, K_{Cs}) \rangle
\end{aligned}
$$

Protocols II: Key establishment and renewal

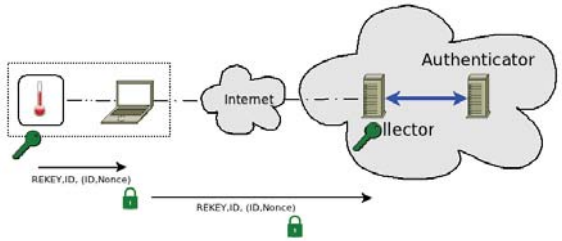


Protocols II: Key establishment and renewal



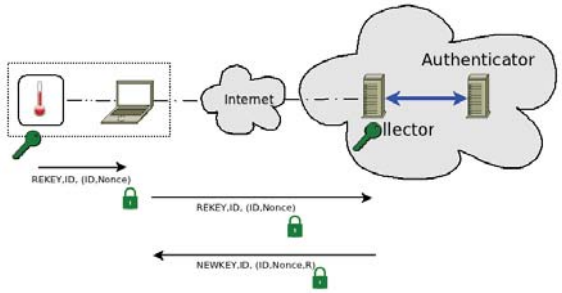
$$s \to S \to C: \quad \langle \text{REKEY}, s, IV, \mathcal{ET}_{Cs}(s, N_s) \rangle$$
$$s \leftarrow S \leftarrow C: \quad \langle \text{NEWKEY}, s, IV, \mathcal{ET}_{Cs}(s, N_s, R) \rangle$$

Protocols II: Key establishment and renewal



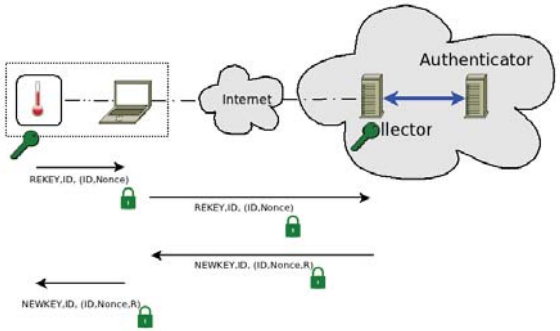
$$s \to S \to C: \quad \langle \text{REKEY}, s, IV, \mathcal{ET}_{Cs}(s, N_s) \rangle$$
$$s \leftarrow S \leftarrow C: \quad \langle \text{NEWKEY}, s, IV, \mathcal{ET}_{Cs}(s, N_s, R) \rangle$$

Protocols II: Key establishment and renewal



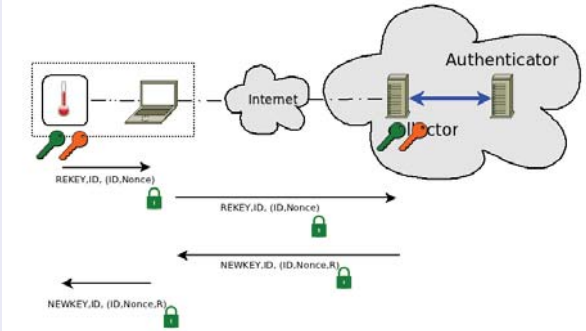
$$s \to S \to C: \quad \langle \text{REKEY}, s, IV, \mathcal{ET}_{Cs}(s, N_s) \rangle$$
$$s \leftarrow S \leftarrow C: \quad \langle \text{NEWKEY}, s, IV, \mathcal{ET}_{Cs}(s, N_s, R) \rangle$$

Protocols II: Key establishment and renewal



$$s \to S \to C: \quad \langle \text{REKEY}, s, IV, \mathcal{ET}_{Cs}(s, N_s) \rangle$$
$$s \leftarrow S \leftarrow C: \quad \langle \text{NEWKEY}, s, IV, \mathcal{ET}_{Cs}(s, N_s, R) \rangle$$

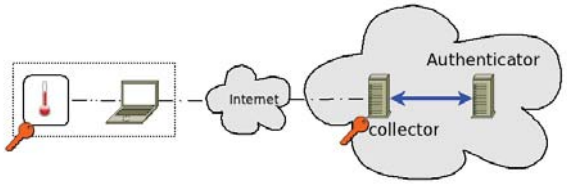
Protocols II: Key establishment and renewal



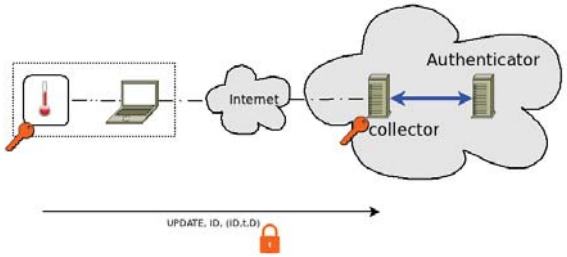
$$s \to S \to C: \quad \langle \text{REKEY}, s, IV, \mathcal{ET}_{Cs}(s, N_s) \rangle$$
$$s \leftarrow S \leftarrow C: \quad \langle \text{NEWKEY}, s, IV, \mathcal{ET}_{Cs}(s, N_s, R) \rangle$$

$$K_T = \text{KDF}(R)$$

Protocols III: Secure data transfer

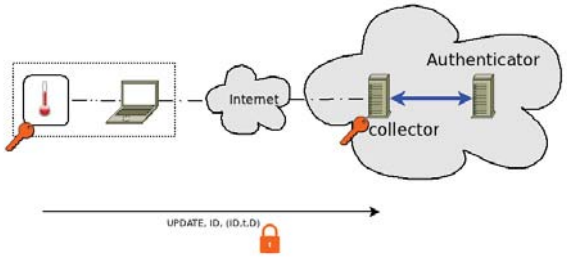


Protocols III: Secure data transfer



$$s \rightarrow S \rightarrow C : \langle \text{UPDATE}, s, IV, \mathcal{ET}_T(s, t, \mathbf{D}) \rangle$$

Protocols III: Secure data transfer



$$s \to S \to C : \langle \text{UPDATE}, s, IV, \mathcal{ET}_T(s, t, \mathbf{D}) \rangle$$

Composition – Encrypt-then-MAC

$$C = \mathcal{E}_T(P) \quad T = \mathcal{T}_T(C) \quad \textit{keypair } K_T = (K_{Te}, K_{Tm})$$
$$m = \langle C, T \rangle$$

Security analysis (informal)

Caveats

- Lack of tamper proofing
- Not formally verified

Security analysis (informal)

Caveats

- Lack of tamper proofing
- Not formally verified

Sensor integrity

Assuming

- Formally specified functionality
- Verified and attested to by a trusted entity
- Per-device unique identity
- Tamper-proofing

Security analysis (informal)

Caveats

- Lack of tamper proofing
- Not formally verified

Sensor integrity

Assuming

- Formally specified functionality
- Verified and attested to by a trusted entity
- Per-device unique identity
- Tamper-proofing

gives

1. Accepted sensor is *a member of the group of trusted devices*

Security analysis (informal)

Caveats

- Lack of tamper proofing
- Not formally verified

Sensor integrity

Assuming

- Formally specified functionality
- Verified and attested to by a trusted entity
- Per-device unique identity
- Tamper-proofing

gives

1. Accepted sensor is *a member of the group of trusted devices*
2. *Graceful degradation* of security in case of sensor compromise

Integrity

Definition

Two sub goals of the integrity objective

1. **Correctness:** *All data received consists of unaltered sensor observations*
2. **Completeness:** *All data produced by trusted sensors is contributed to the final aggregate*

Integrity

Definition

Two sub goals of the integrity objective

1. **Correctness:** *All data received consists of unaltered sensor observations*
2. **Completeness:** *All data produced by trusted sensors is contributed to the final aggregate*

Integrity

Definition

Two sub goals of the integrity objective

1. **Correctness:** *All data received consists of unaltered sensor observations*
2. **Completeness:** *All data produced by trusted sensors is contributed to the final aggregate*

Integrity

Definition

Two sub goals of the integrity objective

1. **Correctness:** *All data received consists of unaltered sensor observations*
2. **Completeness:** *All data produced by trusted sensors is contributed to the final aggregate*

Overall aggregate correctness guaranteed *modulo completeness*

Conclusions

- Correct and **trustworthy** sensing important in networked measurement systems
- Proposal: **trusted sensor** ensures **data integrity**
- **Security guarantees** w.r.t. the correctness sub-goal
- Supported by a **prototype** implementation

Conclusions

- Correct and **trustworthy** sensing important in networked measurement systems
- Proposal: **trusted sensor** ensures **data integrity**
- **Security guarantees** w.r.t. the correctness sub-goal
- Supported by a **prototype** implementation

Future work

- Miniaturization and design of a sensor device
- Tamper-resistance.
- Formal verification procedures
- Public key authentication
- Distributed aggregation