11th International Conference on IT Security Incident Management & IT Forensics (IMF 2018)

PRINCIPLES OF SECURE LOGGING FOR SAFEKEEPING DIGITAL EVIDENCE

Felix Freiling, Friedrich-Alexander-University Erlangen-Nuremberg

Edita Bajramovic, Friedrich-Alexander-University Erlangen-Nuremberg/Framatome GmbH

Hamburg, 09/05/2018

Outline

1 **Overview**

2 **System and Attacker Model**

3 **Security Properties**

4 **Existing Secure Logging Protocols and Their Properties**

5 **Findings and Conclusion**

Overview

- **A log is a “regular or systematic record of incidents or observations”**
- **Logging systems are an integral part of modern server systems**

- **Several secure logging protocols have been proposed:**
  - M. Bellare and B. Yee - Forward integrity for secure audit logs (1997)
  - B. Schneier and J. Kelsey - Secure audit logs to support computer forensics (1999)
  - J. E. Holt - Logcrypt: Forward security and public verification for secure audit logs (2006)
  - D. Ma and G. Tsudik - A new approach to secure logging (2009)
  - R. Accorsi - Bbox: A distributed secure log architecture (2010)

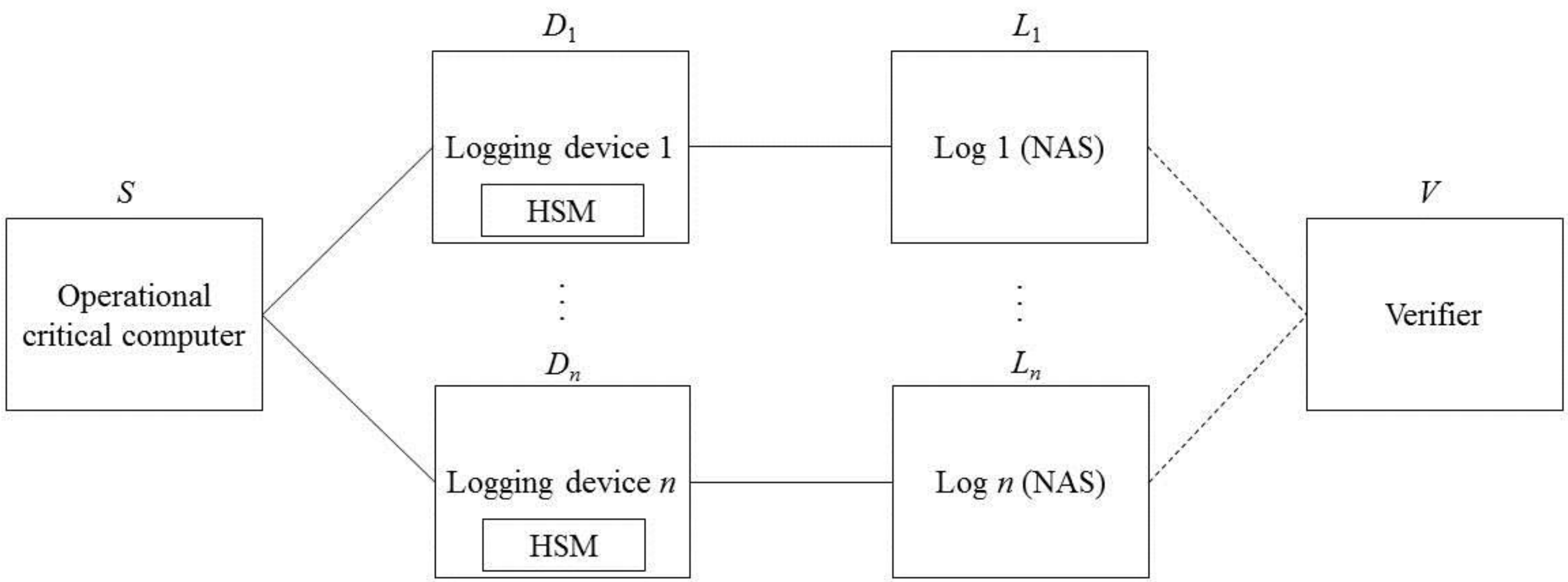
Overview

- **Our aim is to establish a framework to**
  - compare secure logging approaches including their fundamental properties authenticity and completeness
  - identify combinations of assumptions under which it is impossible to implement authenticity or completeness
  - show the precise influence of trusted hardware on the properties that secure logging protocols can achieve

Outline

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

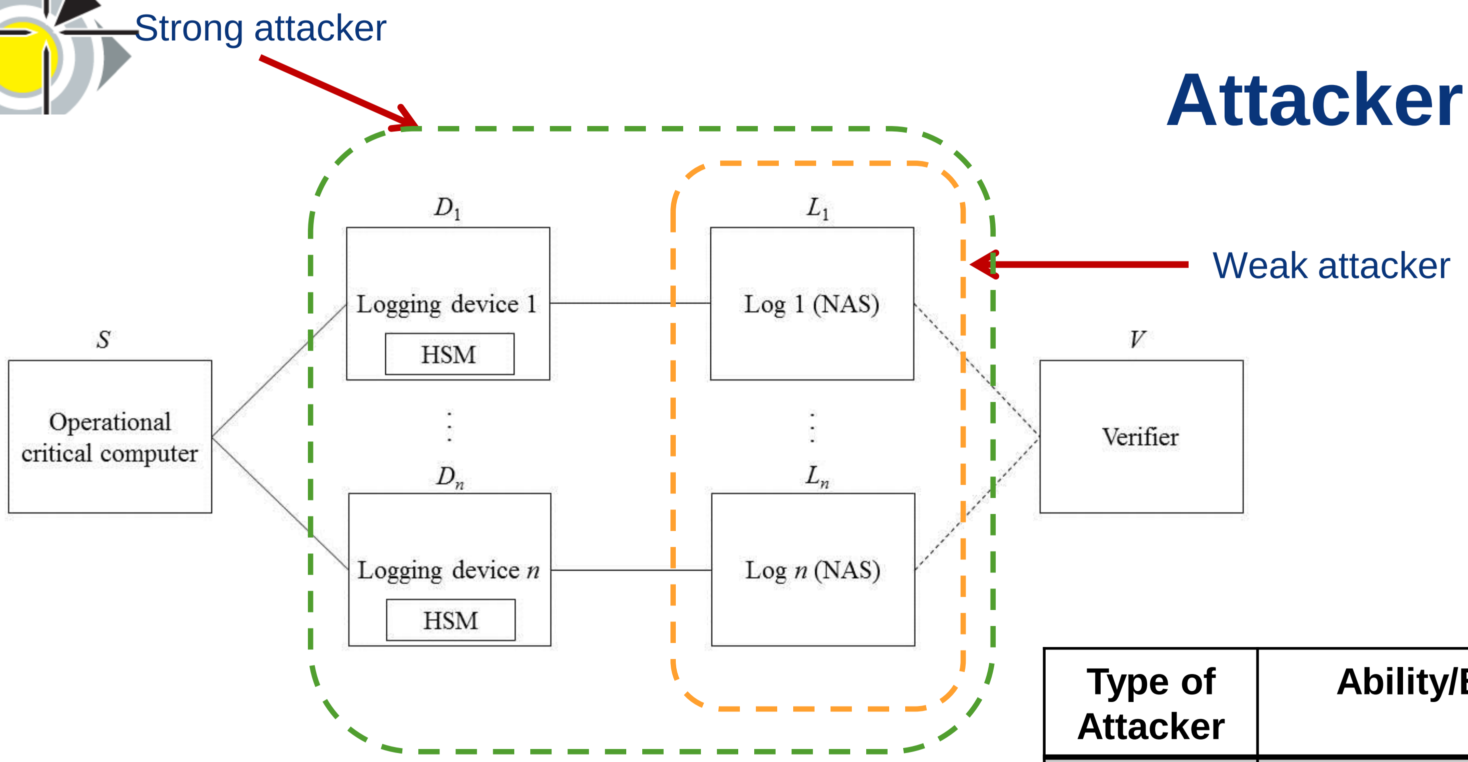
Overall System Model



NAS = Network-Attached Storage

HSM = Hardware Security Mechanism

Attacker Model



Type of Attacker	Ability/Behavior
Local weak	read/write to L_k
Global weak	read/write to $L_1 \ldots\ldots L_n$
Local strong	inject $L_k \Leftrightarrow D_k$ and change D_k
Global strong	inject $L_1 \ldots L_n \Leftrightarrow D_1 \ldots D_n$ and change $D_1 \ldots D_n$

We allow for the existence of a (minimal) tamper proof hardware security module (HSM) within a logging device

Outline

1 Overview

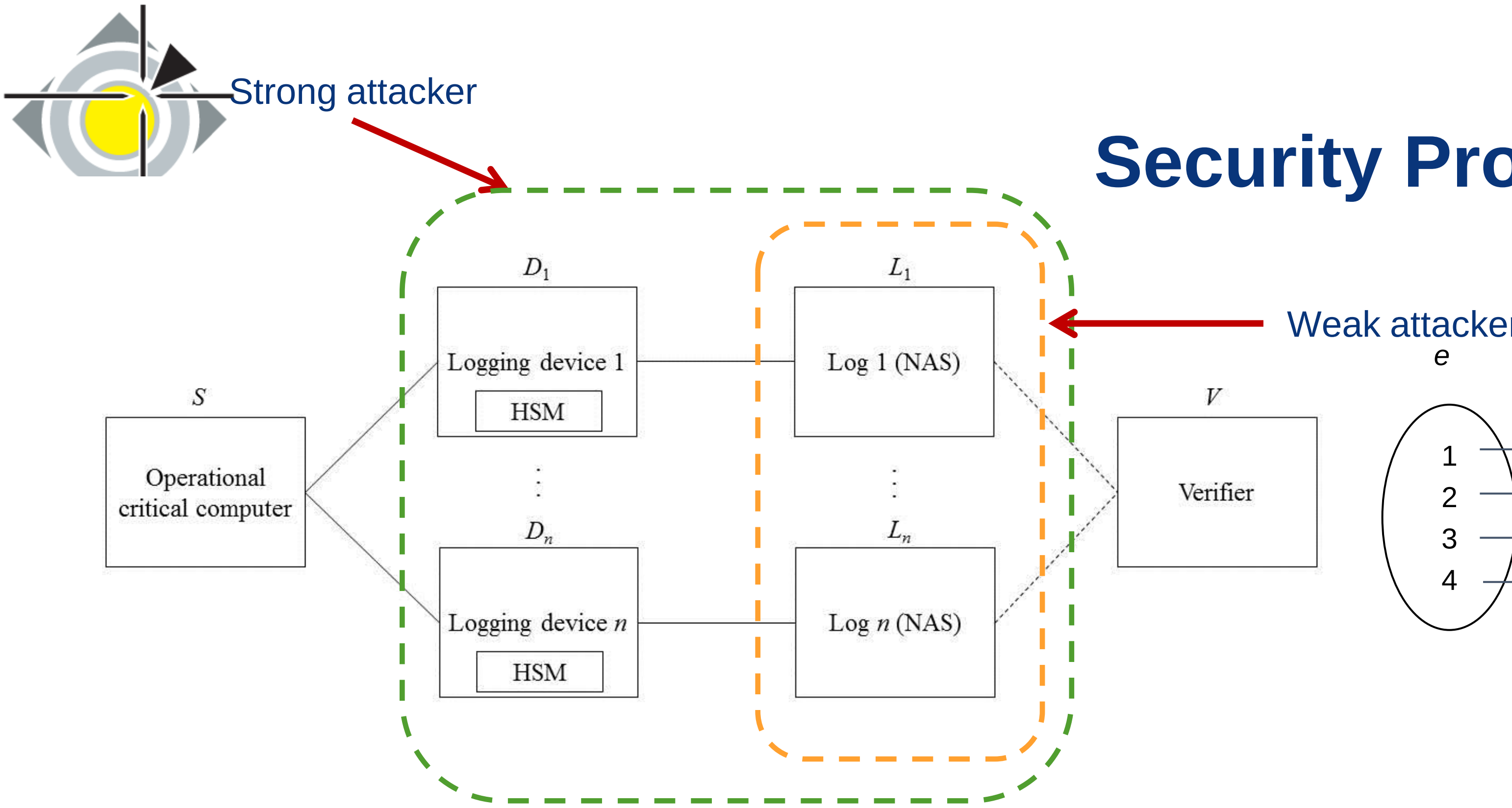
2 System and Attacker Model

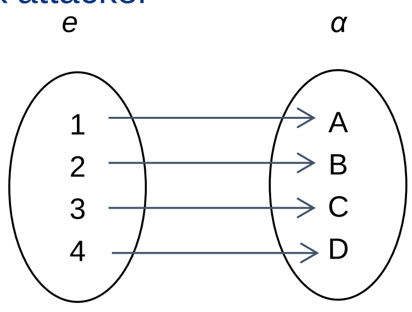
3 **Security Properties**

4 Existing Secure Logging Protocols and Their Properties

5 Findings and Conclusion

Security Properties





- ***AUTHENTICITY*** **– logs are created only if a corresponding event happened**
- ***COMPLETENESS*** **– if (at least after a certain time) every event that happens is actually reflected in the log**
- ***FORWARD-INTEGRITY*** **– successful key compromise only affects a constant number of log entries in the past**

Authenticity, Completeness, and Forward-Integrity

- **Local authenticity (for one log)**
  - a logging protocol satisfies local authenticity for log L_k if and only if for each entry in the log of L_k that is accepted by the verifier, there exists a corresponding event that actually happened
- **Local completeness (for one log)**
  - a logging protocol satisfies local completeness for log L_k if and only if for every event that actually happens, a corresponding log entry eventually exists permanently in the log of L_k which is accepted by the verifier
- **Forward-integrity**
  - a “finite” version of local authenticity
  - authenticity of L_k holds for all log entries that were generated before the attack on L_k took effect

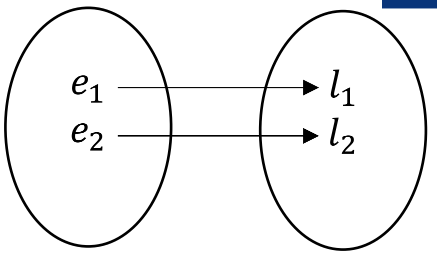
FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Authenticity and Completeness

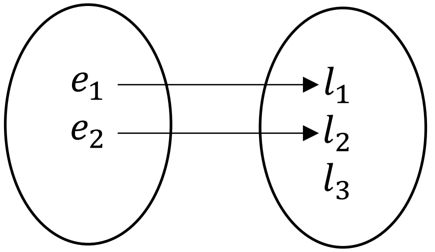
- **Global authenticity (for all logs)**
  - every log in the system satisfies local authenticity
- **Partial local authenticity**
  - a logging protocol satisfies partial local authenticity if and only if it does not satisfy global authenticity but at least one log satisfies local authenticity
- **Global completeness (for all logs)**
  - every log in the system satisfies local completeness.
- **Partial local completeness**
  - a logging protocol satisfies partial local completeness if and only if it does not satisfy global completeness but local completeness for at least one log

Examples and Counterexamples for Authenticity and Completeness

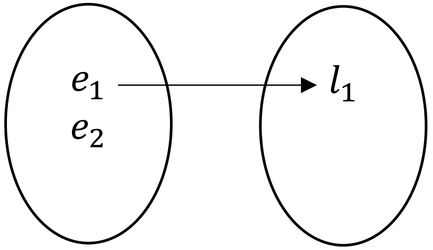
- **A log is complete and authentic**



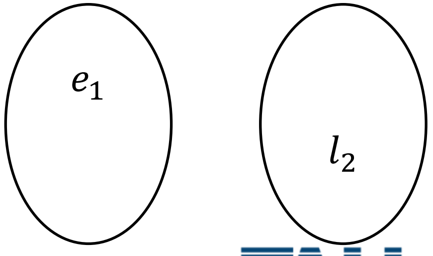
- **A log is complete but not authentic**



- **A log is authentic but not complete**

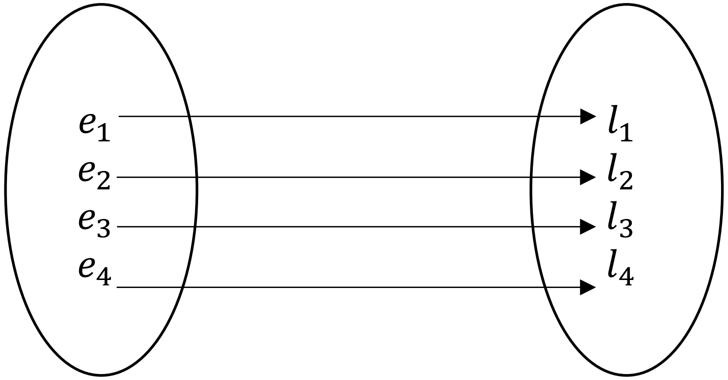


- **A log is neither authentic nor complete**



Outline

Syslog

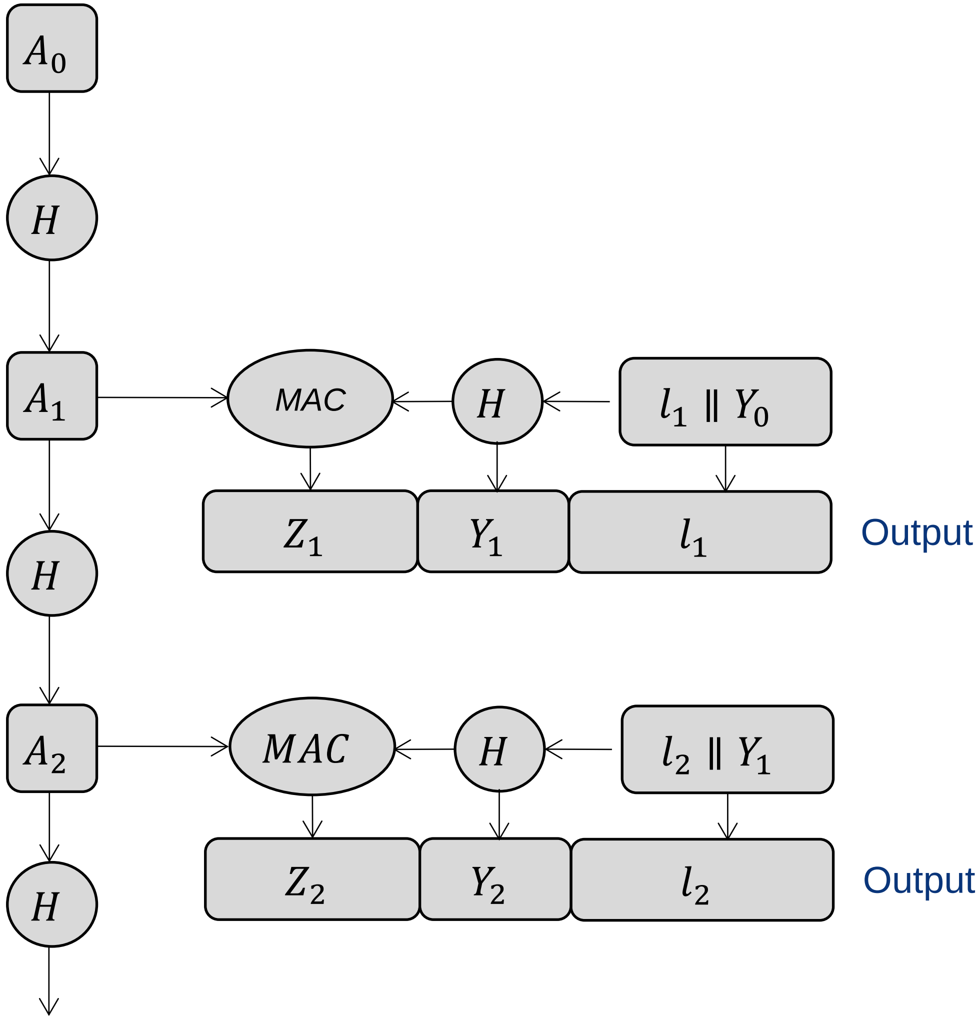


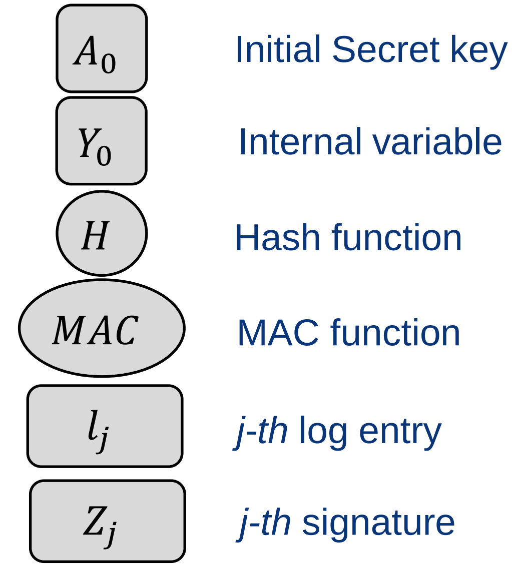
$L_k.append\ (l_j)$

- **Standard syslog event messages are unsigned**
- **A weak attacker is able to add, modify, forge, and delete messages**

- **Achieved Properties**
  - neither authenticity nor completeness against even weak attackers

Schneier and Kelsey





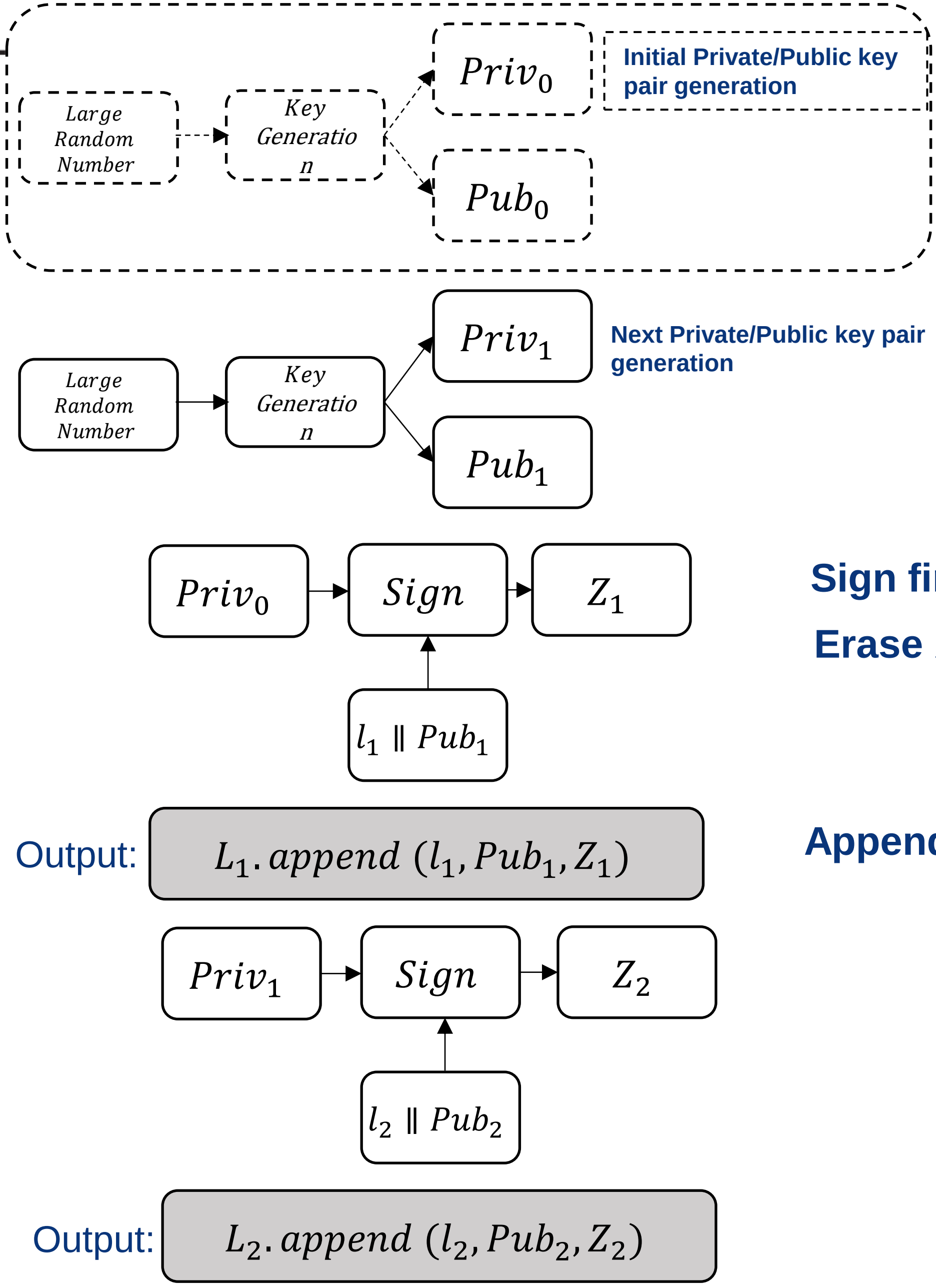
Schneier and Kelsey

- **Original protocol within our framework:**
  - Upon receipt of event message e_j, logging device D_k
    - $Computes\ Y_j \leftarrow H\left(l_j \parallel Y_{j-1}\right)\ for\ l_j \sim e_j$
    - $Computes\ Z_j \longleftarrow MAC(A_{j-1}^k, Y_j)$
    - $A_j^k \leftarrow H(A_{j-1}^k)$
    - $Erases\ A_{j-1}^k\ securely\ from\ memory$
    - $L_k.append\ (l_j, Y_j, Z_j)$

- **Achieved Properties**
  - forward-integrity
  - partial local completeness

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Holt's Logcrypt



Sign first log entry

Erase $Priv_0$ securely from memory

Append (l_1, Pub_1, Z_1) to Log

Holt's Logcrypt

- **Original protocol within our framework:**
  - Upon receipt of event message e_j, logging device D_k
    - $Creates\ a\ new\ public\ private\ key\ pair (Pub_j^k, Priv_j^k)$
    - $Computes\ Z_j \longleftarrow sign(Priv_{j-1}^k, l_j \parallel Pub_j^k)\ for\ l_j \sim e_j$
    - $Erases\ Priv_{j-1}^k\ securely\ from\ memory$
    - $L_k.append\ (l_j, Pub_j, Z_j)$

- **Achieved Properties**
  - forward-integrity
  - global authenticity
  - partial local completeness

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Ma and Tsudik

- **Achieves protection against truncation attacks by replacing one single aggregated signature in the log with every new log entry**
- **Original protocol within our framework:**
  - Upon receipt of event message $\boldsymbol{e_j}$, logging device $\boldsymbol{D_k}$
    - $Computes\ Z_j \longleftarrow MAC(A^k_{j-1}, l_j)\ for\ l_j \sim e_j$
    - $Computes\ Y_j \leftarrow H\left(l_j \parallel Y_{j-1}\right)\ for\ l_j \sim e_j$
    - $A^k_j \leftarrow H(A^k_{j-1})$
    - $Erases\ A^k_{j-1}\ securely\ from\ memory$
    - $L_k.append\ \left(l_j\right)$
    - $L_k.update\ \left(Y_j, 0\right)$
- **Achieved Properties**
  - forward-integrity
  - global authenticity
  - partial local completeness

Outline

1 Overview

2 System and Attacker Model

3 Security Properties

4 Existing Secure Logging Protocols and Their Properties

5 **Findings and Conclusion**

Findings

- **Regarding completeness:**
  - partial local completeness is the best we can achieve in the setting we consider
- **Regarding authenticity:**
  - standard syslog does not achieve any security properties
  - other revisited protocols achieve global authenticity and forward integrity

Conclusion

- We presented a framework in which we could uniformly present the major secure logging approaches, thereby making the comparable
- We were able to show that Schneier and Kelsey and Holt are optimal with respect to achievable security properties
- The problems of truncation attacks were demonstrated by considering the protocol of Ma and Tsudik
- In future, we intend to expand our work in this direction and focus on using consistency conditions to detect log manipulations

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Acknowledgement

Some of the ideas are being elaborated as part of Framatome’s participation in the “SMARTEST” Cybersecurity Testing R&D with three German University partners, partially funded by German Ministry BMWi. We thank the anonymous reviewers for their helpful feedback.

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

11th International Conference on IT Security Incident Management & IT Forensics (IMF 2018)

PRINCIPLES OF SECURE LOGGING FOR SAFEKEEPING DIGITAL EVIDENCE

Thank you for your attention!

Felix Freiling, Friedrich-Alexander-University Erlangen-Nuremberg
Edita Bajramovic, Friedrich-Alexander-University Erlangen-Nuremberg/Framatome GmbH

Hamburg, 09/05/2018