ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΑΡΧΗ ΔΙΑΣΦΑΛΙΣΗΣ ΤΟΥ
ΑΠΟΡΡΗΤΟΥ ΤΩΝ ΕΠΙΚΟΙΝΩΝΙΩΝ

A Framework for Secure and Verifiable Logging in Public Communication Networks

Vassilios Stathopoulos, Panayiotis Kotzanikolaou and Emmanouil Magkos

{v.stathopoulos, p.kotzanikolaou}@adae.gr

emagos@ionio.gr

Hellenic Authority for the Assurance of Communications Security and Privacy – ADAE

(www.adae.gr)

Presentation Structure

- Introduction and Problem Description
- Related Work
- Existing and Proposed Threat Models
- Our Solution
- Conclusions

1. Introduction

- Examined Problem: Secure and Verifiable Logging for Public Network **Providers**
  - ✓ Telephony
  - ✓ ISPs
  - ✓ VoIP, etc
- Public Networks are part of the Critical Infrastructure
- ***Regulatory Authorities*** (***RA***s) are usually responsible to regulate, audit and verify the security level of the Providers

Security Threats

➢ Although Security Standards and measures exist, there are still security holes such as:

- ✓ Intrusion attacks
- ✓ Communication interception
- ✓ Unauthorized access to privileged data (e.g. CDRs)
- ✓ Abuse of privileges

Existing Threat Models

➢ *Trusted Log Generators and Marginally Trusted Log Server*

- ✓ The Logs are generated in a trusted environment
- ✓ The Log Server is marginally trusted and needs to be protected
- ✓ Mainly examines disclosure and modification attacks against the stored logs

➢ *Distributed Log Generators and Marginally Trusted Log Server*

- ✓ In addition to the previous model, attacks during the transmission of log data are considered, e.g.
  - ▪ *impersonation attacks against log generators,*
  - ▪ *disclosure attacks against log messages during transmission*

Our Motivation: The Greek Tapping Scandal

- Victims: VIP Subscribers of a Mobile Telecom Provider

What happen:

- Part of the core network of the provider was compromised by some unknown trojan-like program
- It activated the **LI** (Lawfull Interception) component, already existing in inactive mode
- Turned off the logging of the LI sub-system
- It was discovered after several months period when the trojan affected the SMS subsystem of a roaming provider

The problem

With or without the provider's being informed

- *Log generators* may intentionally be programmed to send modified log messages towards log servers
- *Log Files* may intentionally be modified after their storage to the Log servers.

Our Threat Model

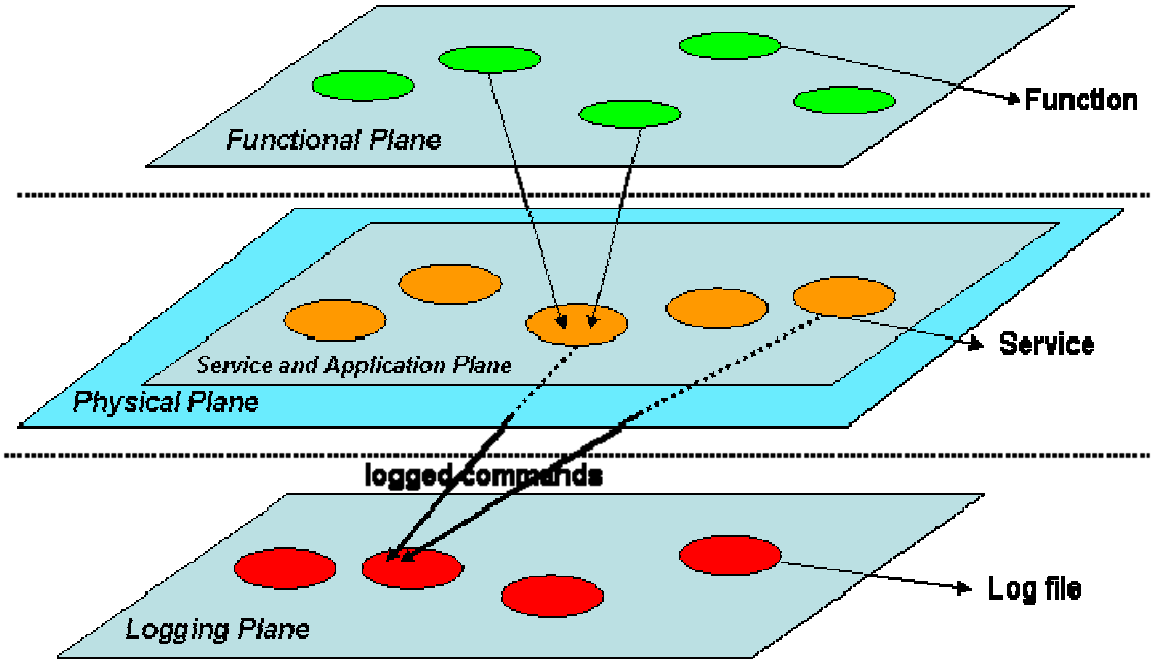
- An extension of the previous models:
- It assumes *Marginally Trusted Log Generators, Log Servers and Communication Channel*
- We consider both ***external*** and ***internal*** attacks
- We also consider attacks such as:
  - ✓ Modification attacks on the stored logs from compromised log servers
  - ✓ Modification attacks from compromised log generators
  - ✓ Modification attacks from cooperating log generators and log servers

Our Solution: A Framework for Secure and Verifiable Logging

- It assumes the partially trusted threat model
- Both external and internal attacks
- A trusted ***RA*** is responsible to audit the network and verify the validity of the log files.
- It consists of 6 phases

Administrative approach

Phase 1: Define a ***Log Reference Model***



It links ***Functions*** *to the corresponding* ***Log Files*** *that monitor these Functions.*

Services *implement the Functions.*

Functional Plane.

It models the **network and operational events** within a network, without taking into consideration implementation details, architectural or topology constraints.

(1) *Security Functions* (*e.g.* system access control, password management, user management, Lawful Interception, Data Retention).
(2) *Service Management Functions* (*e.g*. monitoring, troubleshooting, management services) and
(3) *Network Management Functions* (*e.g.* network configuration, network connectivity, routing).

Service and Application Plane.

- It describes all specific services which are executed within the network or IT nodes.

- It discriminates system from application services, while it takes into consideration the OS platform, communication protocols and interconnections and hardware.

- Examples of services of this plane are the ***snmp service***, the ***dsl service***, the ***password management service***, the ***AAA service***, the ***radius service*** etc.

Logging Plane

- It describes specific ***commands and events*** of each used service, which can be grouped into separated log files.

- For example, the command *"show user"* (captured for displaying user names) will be logged in a *log file* named *"password management".*

- This log file will correspond to the password management service, which implements part of the security management functions.

Phase 2: Define the requirements of log files

- It defines the requirements of each log file regarding:
  - The Structure of each log file
  - Generation frequency of each log file
  - Storage requirements (time, place, format, …)

Technical approach:

Phase 3: Security Measures against External Attacks

- Use the Schneier – Kelsey model
- Uses symmetric cryptography and hash chains
- Update a symmetric key to protect each log entry
- Provides forward security

Schneier – Kelsey model

Each log server is supplied with an initial symmetric key A_0.
The key A_0 is updated for each new log entry , through a cryptographic one-way hash function *hash*, *i.e.* $A_i = hash(A_{i-1})$.
The encryption is $E_{K_i}(D_i)$ where $K_i = hash(W_i, A_i)$ and W_i *is the permission mask of the data entry* D_i.

Each log entry L_i contains the hash value
$Y_i = hash(Y_i-1, E_{K_i}(D_i), W_i)$, (for Y_0 a padding value is used), as well as the Message Authentication Code $Z_i = MAC_{A_i}(Y_i)$.

Schneier – Kelsey model

Weakness

Even in this case, it is possible for a compromised log server to modify the log file.

Example:

If an adversary knows key A_i, and has access to the log files, he reconstructs keys A_{i+1},…A_j and the original log entries are replaced by the manipulated log entries.

Phase 4: Security Measures against Internal Attacks

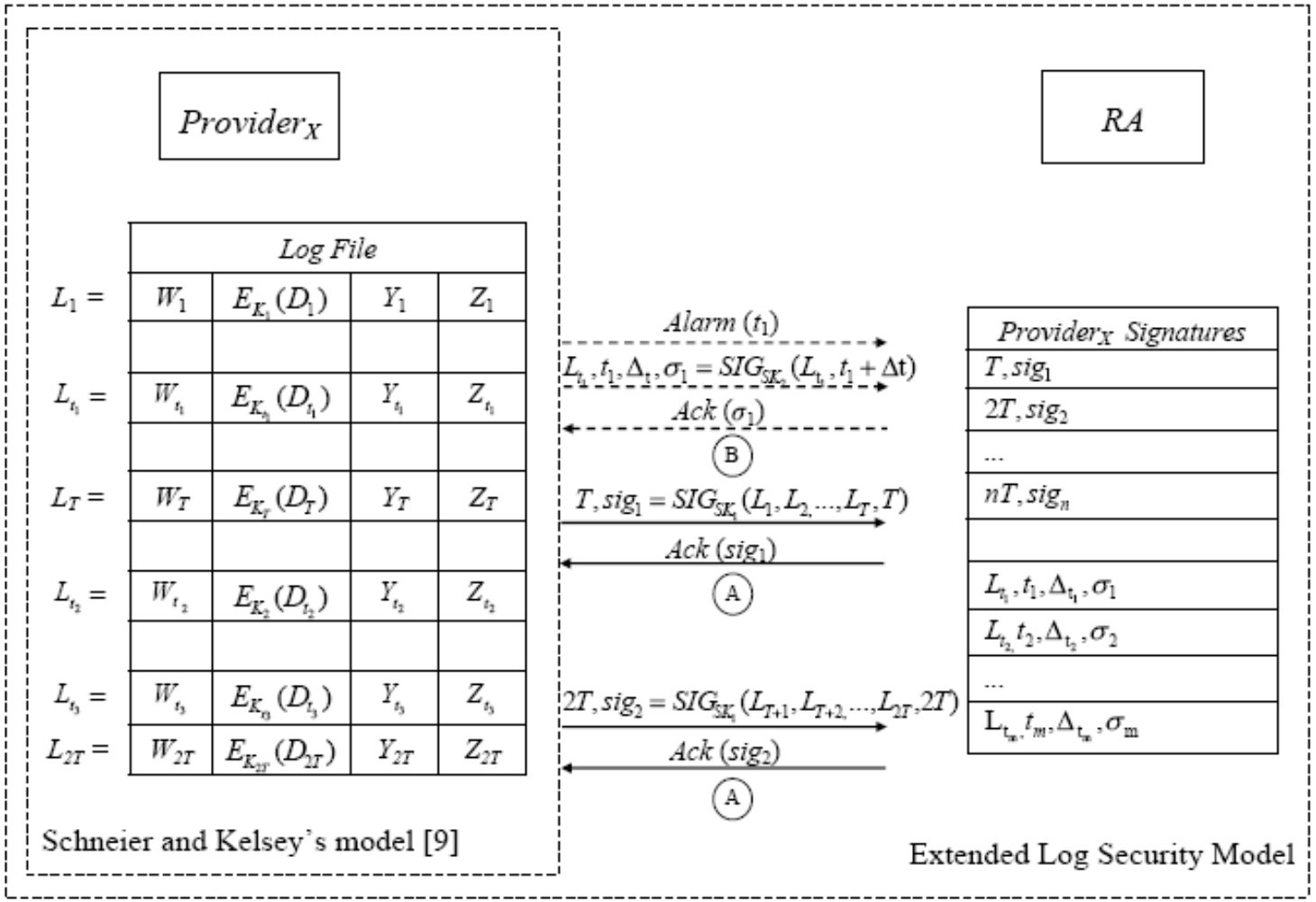
- Extension of Schneier – Kesley's model.
- Limited interaction of the provider with the RA.
- *Manually* generated digital signatures of *Log files* in *predefined periods*
- *Automatically* generated digital signatures of *log entries* in *random* time intervals
- Remote storage of digital signatures in the RA.

Phase 4: Security Measures against Internal Attacks

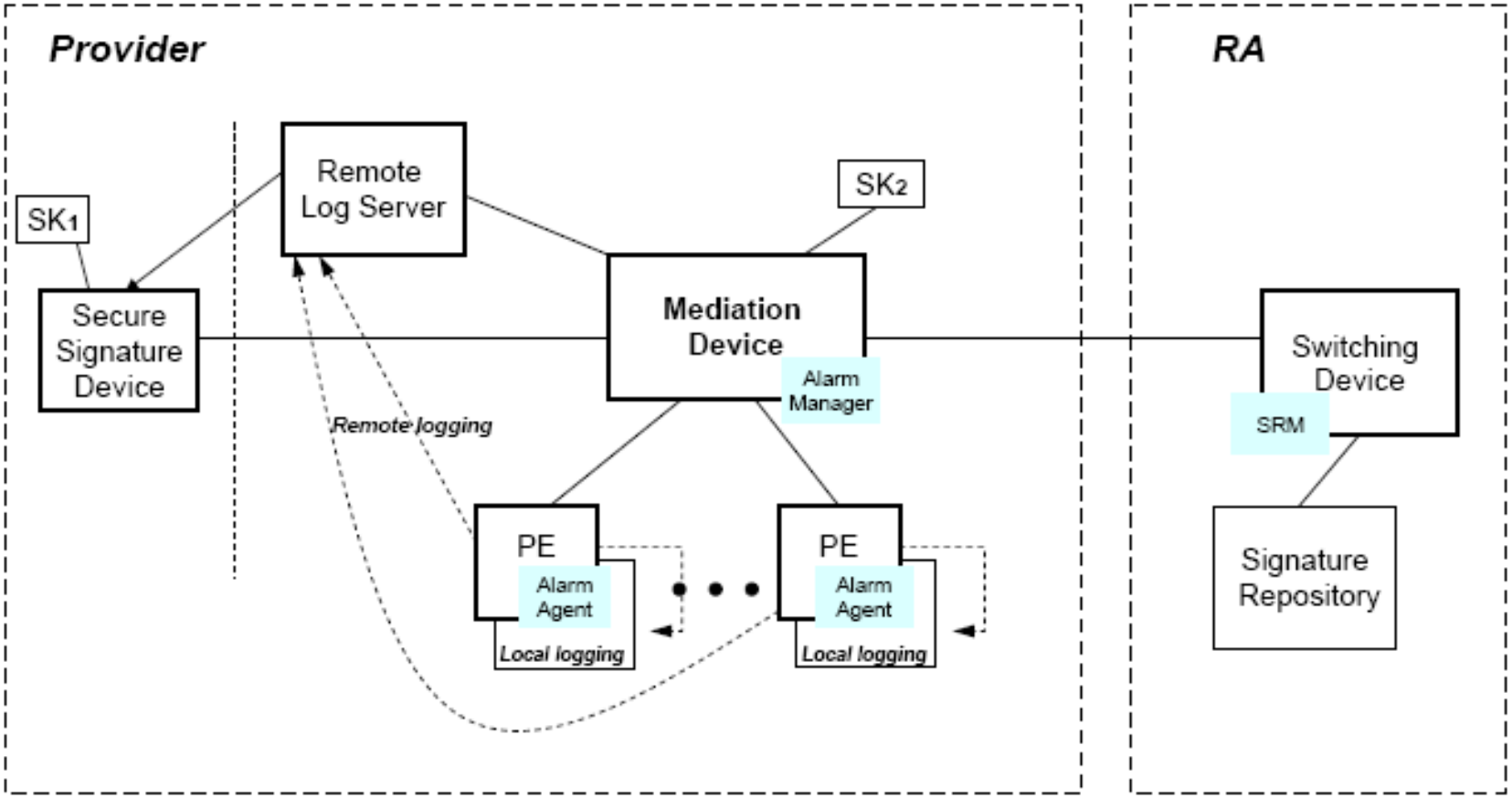
- Manual signatures:
  - ✓ Use a signature key SK_1 in an isolated device.
  - ✓ Only by authorized personnel
  - ✓ The signature period depends on the security and operational needs.
  - ✓ The period is jointly defined by the Provider and the RA.
  - ✓ The signature is used to prove the validity of the Log file in a given time period.

Phase 4: Security Measures against Internal Attacks

- Automated signatures:
  - ✓ Use an independent signature key ***SK_2*** installed in the Mediation Device of the Provider.
  - ✓ It is automatically generated when a ***critical event*** takes place.
  - ✓ The list of critical events is pre-defined by the Provider and the RA.
  - ✓ It may also be triggered by the RA in ***random*** times.
  - ✓ It refers only to a single or a small group of log events, not an entire time period.


Provider_X
RA
Log File
L1 = W1 E_K1(D1) Y1 Z1
L_t1 = W_t1 E_K_t1(D_t1) Y_t1 Z_t1
L_T = W_T E_K_T(D_T) Y_T Z_T
L_t2 = W_t2 E_K_t2(D_t2) Y_t2 Z_t2
L_t3 = W_t3 E_K_t3(D_t3) Y_t3 Z_t3
L_2T = W_2T E_K_2T(D_2T) Y_2T Z_2T
Alarm (t1)
L_t1, t1, Δ_t1, σ1 = SIG_SK_x(L_t1, t1 + Δt)
Ack (σ1)
B
T, sig1 = SIG_SK_x(L1, L2, ..., L_T, T)
Ack (sig1)
A
2T, sig2 = SIG_SK_x(L_T+1, L_T+2, ..., L_2T, 2T)
Ack (sig2)
A
Provider_X Signatures
T, sig1
2T, sig2
...
nT, sig_n
L_t1, t1, Δ_t1, σ1
L_t2, t2, Δ_t2, σ2
...
L_tm, tm, Δ_tm, σm
Schneier and Kelsey's model [9]
Extended Log Security Model

Phase 5: Implementation Design



Phase 6: Log Verification Procedure

1. Verify the manually generated signatures (using PK_1) to verify the validity of the Log file.
2. If the signature is invalid, then the Log file has been tampered by the Provider.
3. Verify the automated signatures (using PK_2) to verify the validity of critical and random log events.
4. If one or more signatures are invalid, then the RA has evidence that at some point the logs have been tampered.
5. If all signatures are verified well, the Provider has evidence that the Log files are secure.

Conclusions

1. Existing Secure Log systems mainly protect from external attacks
2. In Public Communication Networks however, internal or combined attacks must be considered.
3. We extend the Schneier – Kelsey model with:
   - Off-line manual signatures
   - Automated signatures
   - Limited interaction of the RA and periodic remote storage of log file signatures.
4. Operational costs must be considered and minimized.